BIBLIOTECA BASICA
INFORMATICA

INTELIGENCIA ARTIFICIAL Y SISTEMAS EXPERTOS 28



INGELEK

BIBLIOTECA BASICA INFORMATICA

INTELIGENCIA ARTIFICIAL Y SISTEMAS EXPERTOS

28

INGELEK

Director editor:
Antonio M. Ferrer Abelló.

Director de producción:
Vicente Robles.

Coordinador y supervisión técnica:
Enrique Monsalve.

Redactores técnicos:
Juan Lloréns Murillo
Luis Martín Rodríguez

Colaboradores:
Angel Segado.
Patricia Mordini.
Margarita Caffaratto.
Marina Caffaratto.
Casimiro Zaragoza.

Diseño:
Bravo/Lofish.

Dibujos:
José Ochoa.

ISBN del tomo: 84-85831-72-1
ISBN de la obra: 84-85831-31-4
Fotocomposición: Pérez Díaz, S. A.
Imprime: Héroes, S. A.
Depósito legal: M-16948-1986
Precio en Canarias, Ceuta y Melilla: 380 pts.

INDICE

PROLOGO

Desde hace unos pocos meses ha crecido exponencialmente el interés del mundo científico y del campo de la informática por un conjunto de técnicas y teorías conocidas como Inteligencia Artificial. ¿Cuáles son las causas que han motivado el enorme auge de esta nueva disciplina?

Desde luego, el intento de construir máquinas que presenten las capacidades que asociamos con la inteligencia humana no es nada reciente: ya en el siglo XVIII se construyeron los primeros autómatas que intentaban emular aspectos parciales. La causa del relanzamiento de la I.A. provino de la falta de algoritmos matemáticos capaces de enfrentarse a situaciones aparentemente sencillas, tales como el reconocimiento del lenguaje, los problemas de diagnosis (enfermedades, geológicos) o el reconocimiento visual de objetos..

El término Inteligencia Artificial se presta en cierta medida a interpretaciones erróneas y, tal y como ocurrió con los "Cerebros Electrónicos"; quizá dentro de unos pocos años esta denominación haya desaparecido y surjan otros términos más acordes con la realidad de las investigaciones.

Hoy en día los programas más avanzados son más sistemas basados en el conocimiento de expertos que sistemas inteligentes, ya que sus capacidades son, aunque de una cierta potencia, muy limitados en sus aplicaciones, restringiéndose a temas muy concretos. Tales son los ya muy famosos sistemas de diagnosis de enfermedades infecciosas o de prospecciones geológicas, que están siendo aplicados con mucho éxito por numerosos hospitales y por las compañías petrolíferas.

El resultado de unir el conocimiento de un experto con los ordenadores más potentes y rápidos está creando un nuevo entorno que cambiará completamente las estrategias de resolución de problemas antiguos y además permitirá enfrentarse a otros intratables en la actualidad. En todos aquellos problemas donde partiendo de unas combinaciones iniciales el sistema se expanda de una forma exponencial, el ordenador, por rápido y potente que fuera, nunca iría por delante de la creciente complejidad del modelo y, como resultado, las soluciones siempre serían parciales. Incorporar un conocimiento, una heurística, que permita al programa eliminar aquellos caminos cuya posibilidad de ser correctos es más baja, facilitará la resolución de estas situaciones e impulsará el esclarecimiento de numerosas cuestiones.

La Inteligencia Artificial está siendo considerada por numerosos países uno de los aspectos clave del progreso en los próximos años. Dentro de estos países hay que citar, por su importancia, el Japón, que ha incluido de manera preferente los tres campos de la I.A. (sistemas expertos, robótica y lenguaje natural) en el conjunto de temas a investigar dentro de los planes de la quinta generación de ordenadores.

Es, por tanto, importante que el lector vaya teniendo una idea general sobre las perspectivas y limitaciones que nacen con la irrupción de la Inteligencia Artificial en el sector informático en general y pueda iniciarse en otra de las ramas de conocimiento que se abren a nuestros ojos.

CAPITULO I

INTRODUCCION A LA INTELIGENCIA ARTIFICIAL

Desde el inicio de la era de los ordenadores, los especialistas en informática han tratado de desarrollar técnicas que permitan a las computadoras actuar como lo hace el ser humano. Una de las bases de apoyo de esta nueva forma de diseñar un programa es la Inteligencia Artificial.

Se conoce como Inteligencia Artificial una nueva forma de resolver problemas dentro de los cuales se incluyen los sistemas expertos, el manejo y control de robots y los procesadores de lenguaje natural.

Un conjunto de las técnicas de la I.A. se están aplicando recientemente en la construcción de sistemas expertos, que permiten a los ordenadores ayudar al hombre a resolver problemas y en la toma de decisiones.

Esta nueva forma de enfrentarse a los problemas va a producir en los próximos años una revolución dentro de la naciente relación entre el usuario y el ordenador, comparable a la introducción de los microordenadores en todas las oficinas y hogares. Por una parte ayudará a la resolución de situaciones que antes, por su complejidad, eran casi imposibles de tratar mediante la programación tradicional, y por otra, racionalizará y clarificará la información accesible al usuario, es decir, resumirá y simplificará la información, destacando sólo aquella que es realmente importante para las necesidades del usuario.

Considere, por ejemplo, el problema de diseñar un programa de entrenamiento para el personal de ventas de una empresa, donde los productos y servicios de cara al cliente están cambiando constantemente. Se puede enseñar la técnica de ventas en general, pero los aspectos específicos de cada producto seguramente

habrán cambiado al tiempo de terminar el curso. Imagine en lugar de esto un programa capaz de interaccionar con el vendedor haciéndole preguntas y recomendándole las opciones apropiadas: sería igual que tener para cada vendedor un especialista puesto al día constantemente sobre los nuevos productos. El programa contará con un sistema asequible y sencillo que permita modificar la información que maneja y renovarla constantemente; además, el cambio del programa será realizado por los especialistas del producto en la empresa y no será necesaria la intervención de un programador de ordenadores. Este es un ejemplo posible de lo que llegará en los próximos años.

Pronto aparecerán también puestos de información inteligentes dentro de cada departamento de las fábricas, capaces de manejar más datos y de forma más compleja que los actuales ordenadores. La utilización de estos terminales por parte de los jefes de departamento o sección les permitirá controlar más actividades y de una forma más ordenada, lo que redundará en el incremento de la cantidad y calidad de las decisiones que pueda tomar.

Estas posibilidades que se vislumbran están llevando a la mayoría de las grandes empresas de todo el mundo a crear departamentos de investigación en el campo de la I.A. y a rescatar de las universidades el conocimiento sobre los sistemas expertos que éstas poseen y que han venido elaborando desde la década de los años sesenta. El salto de las universidades a la industria lleva consigo que el mercado de las aplicaciones de I.A. esté empezando a expandirse; los primeros resultados prácticos se están comercializando ya en los países más avanzados (Estados Unidos, Japón, Francia e Inglaterra).

La génesis de la Inteligencia Artificial

A finales de la segunda guerra mundial aparecieron en escena los primeros grupos de investigación que intentaban desarrollar una máquina capaz de manejar de forma automática datos y números y llevar a cabo complejas operaciones matemáticas. Dentro de ellos se podían distinguir dos tendencias: por una parte, aquellos que pensaban que las instrucciones fundamentales de funcionamiento de una máquina de tales características debían ser los operadores lógicos "AND", "OR", "NOT", y, por otra, aquellos que veían mucho más ventajoso (aunque menos potente en sus realizaciones) utilizar los operadores numéricos "+", "×", ">",...

En aquellos tiempos la lentitud y capacidad de los primitivos ordenadores hizo que se diseñaran y construyeran en base a los operadores matemáticos, que poseen una capacidad de manejar

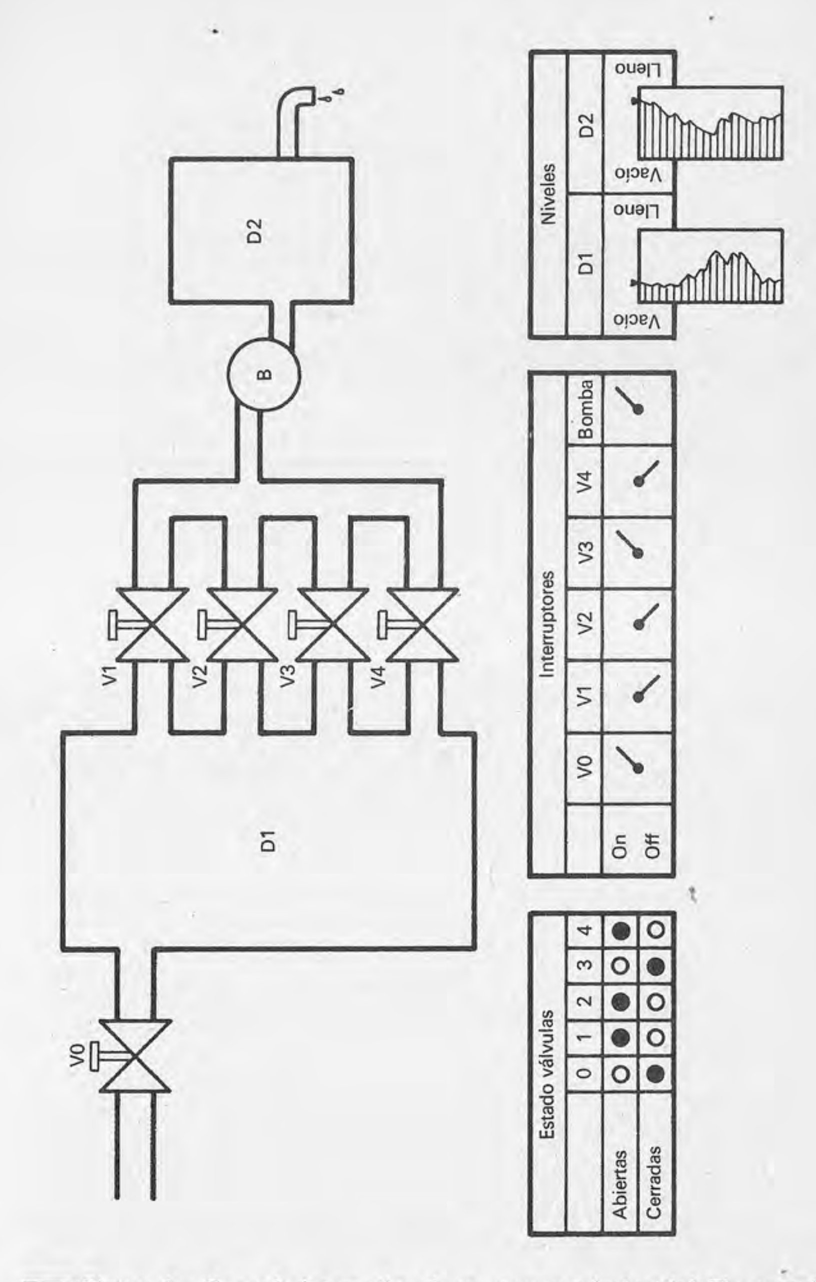


Figura 1.—Consola interactiva de un proceso industrial. El operario puede modificar e intervenir desde la consola en el proceso que controla.

números muy buena, aunque fallan bastante en el manejo de la lógica.

Con el paso de los años los grupos de investigación basados en los operadores lógicos fueron abriéndose camino dentro de las universidades y dando los primeros pasos en lo que hoy llamamos I.A. El camino fue largo y costoso, ya que los resultados obtenidos consistían en sistemas lentos en la ejecución y el producto final no era lo suficientemente práctico. Finalmente, y debido a un factor externo como es el desarrollo de la microelectrónica, que impulsó la construcción de una nueva generación de ordenadores de mayor capacidad, y a su rapidez y menor coste, se ha provisto a la Inteligencia Artificial del hardware apropiado a sus necesidades.

Tradicionalmente se divide la Inteligencia Artificial en tres grandes aplicaciones: los procesados de lenguaje natural que facilitan la comunicación del ordenador con el usuario, la robótica y todo lo asociado con la visión y manipulación de objetos y los sistemas expertos, basados en el almacenamiento del conocimiento de un experto.

Se puede apreciar en la figura 2 la cronología y los distintos factores que han impulsado las técnicas de la I.A. y los resultados que se esperan en los próximos años.

Todos estos factores que se han comentado anteriormente no hubieran podido, sin embargo, atraer la atención de las grandes empresas y del mercado informático en general si no fuera por la aparición de nuevos retos a los que se están enfrentando las empresas del sector informático. Uno de estos retos, y quizá el más importante, es la simulación de circunstancias y problemas reales.

La simulación en un ordenador de una situación real, por pequeña y sencilla que parezca, sólo es posible cuando las soluciones o estructuras en las que se expande el problema son pocas; a este tipo de problemas se les conoce como de explosión combinatoria. En aquellos casos donde las combinaciones son muchas, el ordenador, por grande y rápido que sea, nunca superará la complejidad del modelo.

La solución no es, como se intuye, hacer más potente al ordenador. En vez de generar todas las soluciones o estructuras posibles, el programa sólo generará aquellas cuya posibilidad de ser ciertas es más alta. Consideremos el caso de un problema de ajedrez. ¿Cómo es posible que con todos los adelantos producidos, un programa de ajedrez no sea capaz de ganar siempre a un gran maestro? La solución es bien sencilla: el hombre, que no posee ni la rapidez ni la memoria de una computadora, posee, sin embargo, una capacidad que le destaca; el ajedrecista parte de antemano con una heurística, tiene una experiencia, un conjunto de reglas, que le permiten eliminar de las miles de jugadas posibles

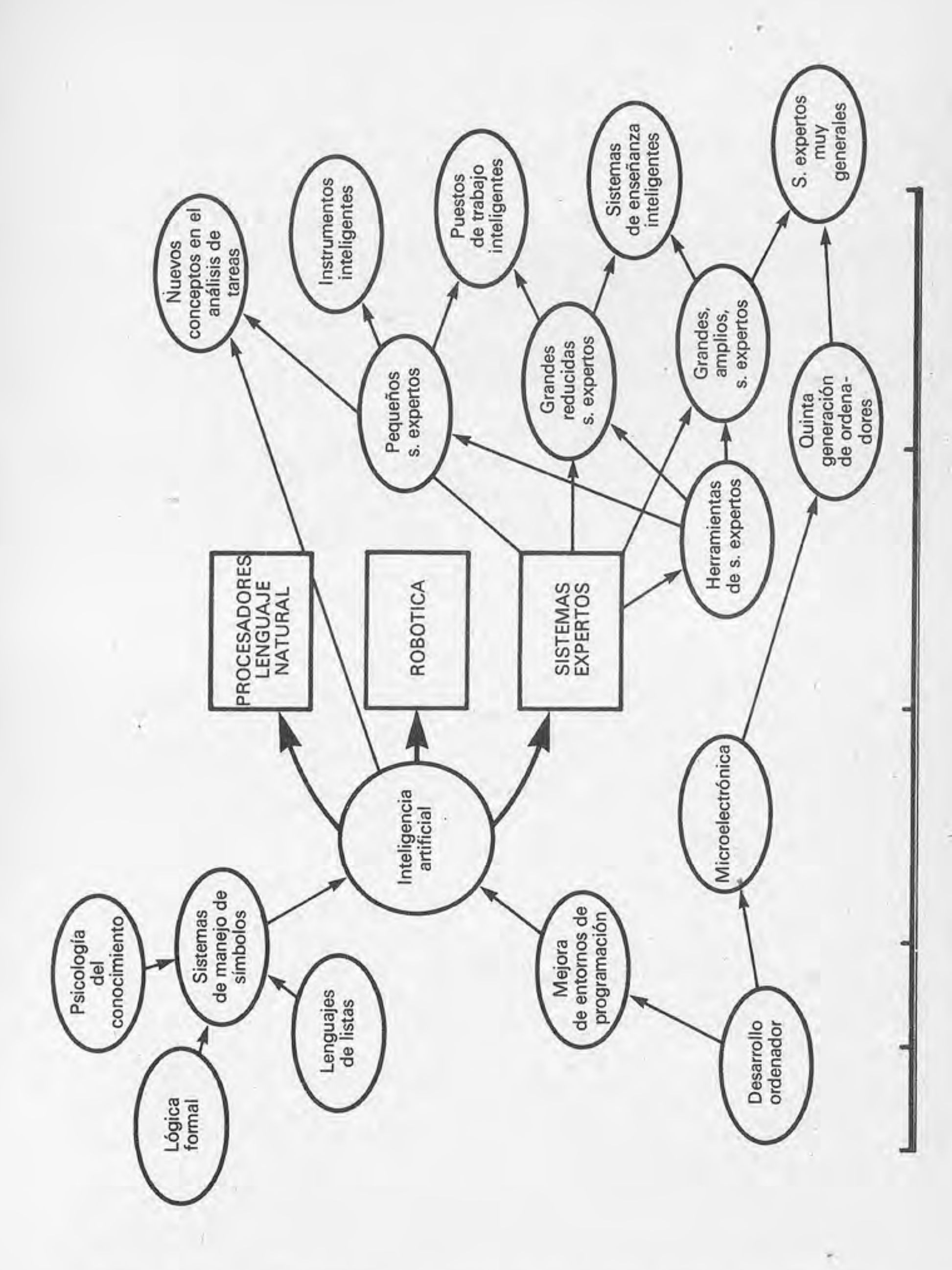


Figura 2.—Evolución de los sistemas expertos, partiendo de la lógica y el conocimiento hasta llegar a los sistemas inteligentes.

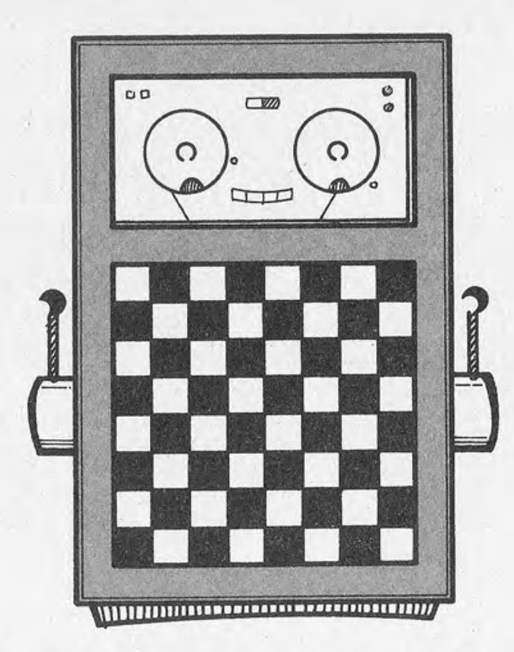


Figura 3.—Una de las últimas aplicaciones de la I.A. son los juegos. No se les da más capacidad, sino más conocimiento.

aquellas cuya posibilidad de ser correctas es más baja, estudiando en detalle sólo un conjunto reducido de las jugadas iniciales.

Este tipo de conocimiento, la experiencia del experto en un tema concreto, la posibilidad de incorporar un conocimiento al programa es lo que ha llevado al ser humano a investigar en el campo de la Inteligencia Artificial, cuyo desarrollo se observa en el cuadro de la figura 4

La I.A. trata del diseño de sistemas informáticos inteligentes, esto es, sistemas que presentan las características asociadas con la inteligencia humana: entendimiento del lenguaje, aprendizaje, razonamiento, resolución de problemas, etc.

Dentro de este concepto han aparecido dos escuelas. Una, llamada de síntesis, que trata de construir sistemas que presenten inteligencia independiente de si usan o no internamente los mismo métodos que el hombre, y otra, conocida como de simulación, que intenta emular la forma en la que el ser humano presenta la facultad de ser inteligente.

Período	Sucesos Claves
Antes segunda guerra mundial	Lógica formal Psicología del conocimiento
La postguerra 1945-1954, inicio de la I. A.	Desarrollo del ordenador Conferencias sobre cibernética
Los años de formación, 1955-1960	Aumenta la disponibilidad de ordenadores Lenguaje de proceso de información (IPL-I) Psicología del proceso de información
Los años del desarrollo, 1961-1970	LISP Heurística Robótica Solución de problemas de ajedrez DENTRAL (Stanford)
La especialización y los sistemas expertos, 1970-1980	MYCIN (Stanford) MACSYMA (MIT) Ingeniería del conocimiento EMYCIN (Stanford) PROLOG
La comercialización, 1981	PROSPECTOR(SRI) Proyecto japonés de la quinta generación INTELLECT (A.I.C.) Sistemas inteligentes de recuperación de información Diversas compañías comercializan herramientas para la construcción de sistemas expertos 

Figura 4.—Aspectos claves en la historia de la Inteligencia Artificial.

Retos a los que se enfrenta la Inteligencia Artificial

De todas las facultades que hacen al ser humano inteligente, los investigadores en I.A. están centrando sus estudios en tres grandes grupos:

• Comunicación y percepción.
  Lenguaje natural.
  Visión.
  Manipulación
• Razonamiento simbólico.
• Ingeniería del conocimiento.

1. Comunicación

Desde el inicio de la década de los cincuenta, cuando por primera vez se intentó construir un sistema de traducción automática mediante la utilización del ordenador, los programas que entienden un lenguaje humano se han convertido en uno de los grandes retos de la informática. Pronto se vio la fuerte relación entre este campo de la I.A. y la lingüística, lo que está ayudando incluso a comprender la naturaleza del lenguaje humano.

Los japoneses, en concreto en su plan de la quinta generación de ordenadores, han incluido a la comunicación dentro de los temas cuyo avance es necesario para el desarrollo de la ciencia; y no sólo los japoneses, sino también los países de la Comunidad Económica Europea y los americanos consideran este campo fundamental como herramienta de progreso. Dentro de estos planes se incluyen tres grandes proyectos:

— Traductor automático multilingüe, con una capacidad de 100.000 palabras.
— Sistemas de consulta sobre diversos temas, con más de 5.000 palabras y 10.000 reglas de inferencia.
— Sistema capaz de hablar y entender el lenguaje natural, con cerca de 10.000 palabras de vocabulario.

Sin embargo, y a pesar de estos proyectos, han aparecido autores que se oponen a la utilización directa del lenguaje natural basándose en que dificultará la comunicación en vez de hacerla más sencilla. Han surgido problemas tales como el análisis sintáctico y semántico de las oraciones, la ambigüedad de palabras con múltiples significados, la determinación del significado de palabras simples donde el contexto general es el que las condiciona, etc. El lenguaje humano, aunque bueno para comunicarnos, es demasiado ambiguo e inexacto, no muy apropiado para transmitir al ordenador instrucciones precisas sobre lo que debe llevar a cabo.

2. Razonamiento Simbólico.

En la informática tradicional los ordenadores trabajan con programas en los cuales los datos acceden directamente al ordena-



Figura 5.—La comunicación entre el hombre y el ordenador en nuestro propio lenguaje es uno de los aspectos clave para los japoneses en su proyecto de quinta generación.

dor, pero las decisiones sobre cómo procesar dichos datos están impresas en el lenguaje del programa y almacenadas en la memoria durante el proceso de ejecución. Por ejemplo, el programa que gestiona las cuentas corrientes en un banco recibe todas las noches unos datos diferentes, pero la forma de procesarlos es invariablemente la misma todos los días.

La Inteligencia Artificial intenta integrar el conocimiento en el

sistema; en otras palabras, un sistema inteligente que escribe su propio programa.

Los sistemas inteligentes se basan en reglas heurísticas, en contraste con los programas de cálculo numéricos, basados en el uso de ecuaciones analíticas. La heurística hace hincapie, dentro del programa, en los aspectos del problema que parecen más críticos y en las partes de la base de conocimiento que parecen más relevantes, y guía al programa en los casos particulares desechando ciertos caminos y centrándose en otros. El resultado es que el programa sigue una línea de razonamiento en vez de seguir una secuencia de pasos fijos en el cálculo.

3. Ingeniería del conocimiento.

Bajo la denominación de ingeniería del conocimiento se agrupan todas las áreas que intervienen en el desarrollo de los sistemas expertos y bases de conocimientos. Dentro de estas áreas aparecen los siguientes campos:

- Representación del conocimiento. Se están desarrollando en la actualidad una gran variedad de métodos, todos ellos para facilitar el razonamiento simbólico y permitir la codificación y aplicación del sentido común. Los esquemas más habituales de representación son las reglas de producción, las estructuras lógicas y el cálculo de predicados.
- Adquisición del conocimiento. Obtener el conocimiento necesario para la creación de un sistema experto no es una tarea sencilla. En algunos temas el sistema puede aprender a través de la experiencia, pero normalmente el experto y el programador del sistema deben trabajar unidos para lograr condensar el saber en unas ciertas reglas lógicas. Actualmente se está trabajando sobre ciertos programas que reciben la sabiduría del experto mediante sesiones de enseñanza. El ordenador va preguntando, analizando las respuestas e incorporándolas a la base en forma de reglas lógicas.
- Métodos de inferencia. Son los métodos que trazan una línea de razonamiento a una pregunta dada. Estas técnicas de generación y análisis de hipótesis son técnicas de concentración, al igual que las heurísticas ya mencionadas, es decir, centran la atención en determinadas reglas y registros, marcando la línea de acción. Un problema asociado a estos métodos es la cuantificación y manejo de datos indeterminados, para el cual, aunque a veces se utiliza el teorema de Bayes o el análisis de decisión, se requiere generalmente el uso de la lógica difusa (Zadeh, 1965).

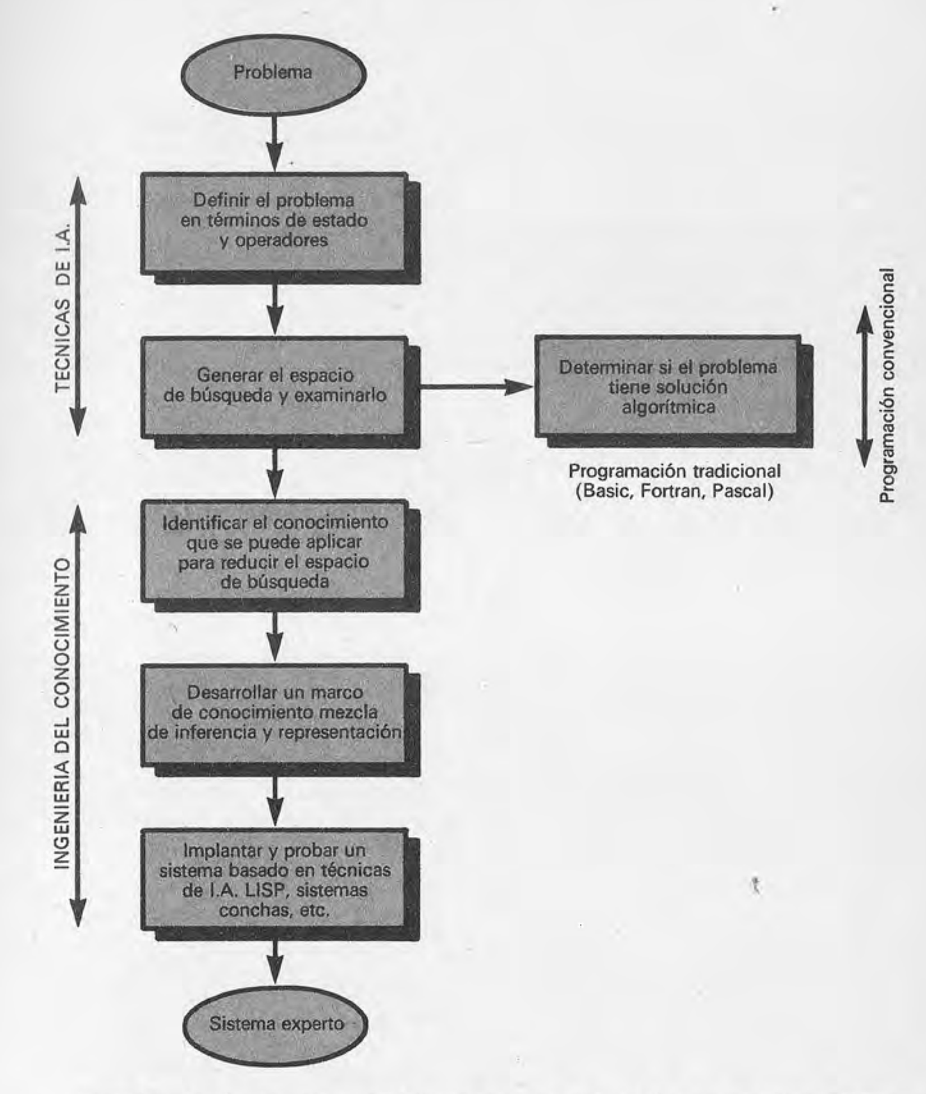


Figura 6.—La definición del problema es uno de los aspectos claves en el desarrollo de los sistemas expertos.

Fronteras de la Inteligencia Artificial

Aunque es difícil en la actualidad predecir lo que serán capaces de hacer los sistemas inteligentes dentro de sus múltiples

aplicaciones, sin embargo, a partir del análisis de lo que es posible y lo que no de los sistemas actuales, se pueden extrapolar algunas ideas a lo que será el futuro de la I.A. en los próximos años.

Hoy día, en que la tecnología no ha hecho nada más que comenzar, las aplicaciones son muy restringidas; los sistemas basados en el conocimiento sólo son capaces de enfrentarse y resolver problemas dentro de un campo muy delimitado. No son capaces de razonar a partir de axiomas o teorías generales y sólo aceptan un tipo de hechos y heurística determinados. No tienen desarrollada la facultad de aprender ni pueden razonar por analogía.

La consecuencia de todos estos factores se resume en dos características:

a) La falta de sentido común.
b) Sus razonamientos se deterioran rápidamente cuando el problema se sale fuera de la tarea específica para la que fue diseñado.

Por otra parte, los sistemas basados en el conocimiento no emiten juicios incoherentes, no se saltan ninguna posibilidad ni se empeñan en mantener una postura en contra de los hechos reales. Tampoco tienen malos días, siempre tienen en cuenta todos los detalles y sistemáticamente consideran todas las posibles alternativas. Los mejores sistemas expertos, que contienen miles de hechos y reglas y sobre todo están diseñados para una función apropiada (como, por ejemplo, toda clase de diagnósticos médicos, geológicos, etc.), realizan las mismas funciones que el mejor experto y sus juicios y conclusiones son equiparables e incluso mejores que la media de especialistas en dicho tema.

CAPITULO II

CONCEPTOS BASICOS

De la Inteligencia Artificial, a los sistemas expertos

Cuando nació la ciencia de la informática se pensó que las nuevas máquinas, capaces de realizar operaciones aritméticas por sí solas y con una cierta capacidad de operación, iban a resolver absolutamente todos los problemas que tenía el hombre.

Ya entonces se empezó a pensar en ordenadores que hablaran en todos los idiomas de la tierra, robots "amas de casa", grandes "androides" (mitad hombre, mitad robot) capaces de trabajar en las peores circunstancias, etc. Sin embargo, esta euforia inicial, causada principalmente por el desconocimiento real de la nueva rama de la ciencia, acabó muy pronto: Un ordenador tan grande como una casa (el ENIAC, por ejemplo) necesitaba "una eternidad de tiempo" para realizar una simple multiplicación.

Posteriormente, cuando pasados unos años de estudios sobre el camino que debía tomar esta ciencia en su evolución se resolvieron los primeros problemas físicos (como el paso de las válvulas a los transistores y de estos últimos a los circuitos integrados —las famosas "cucarachas"), comenzó de nuevo otra euforia sobre la capacidad límite de los nuevos ordenadores. Se podría conseguir un ordenador que fuera capaz de almacenar toda la información que existe en la Biblioteca Nacional de forma que el lector no tuviera que desplazarse al edificio de la Biblioteca, sino que desde su casa podría consultar cualquier libro que estuviera almacenado en la memoria del "maravilloso ordenador". Y no sólo se pensó en estas tareas: los ordenadores serían capaces de sa-

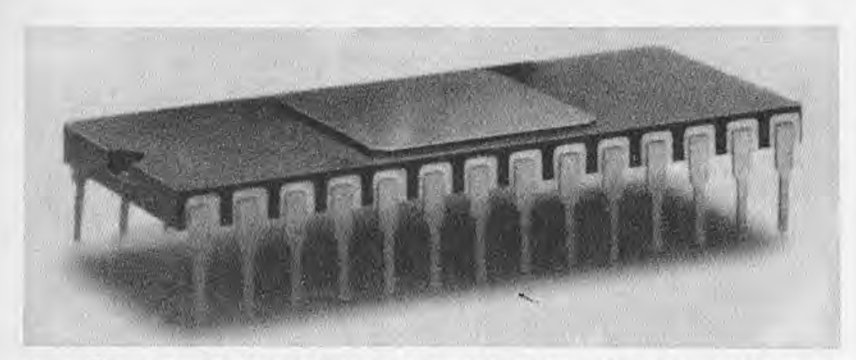


Figura 1.—En los ya famosos "chips" se encuentra el presente (quizá el futuro) de la informática. Su capacidad de operación está creciendo día a día.

berlo todo. Si alguien necesitara saber cuántos goles le metió el Athletic de Bilbao al Castellón en el partido de fútbol de Copa en 1973 y cuántos recibió, bastaría con preguntárselo al ordenador.

De nuevo el tiempo volvió a demostrar a los desmesuradamente optimistas que lo que ellos deseaban todavía iba a tardar muchos años en poder realizarse.

Sin embargo, un "poso" de estas magníficas ideas sí que logró cundir en algunos profesionales de la informática.

Ellos pensaron: "desde luego es muy difícil que un ordenador lo sepa *todo* sobre *todo* lo que ocurre en el mundo. Sin embargo, sí podría ocurrir que un ordenador lo programáramos para que supiera todo pero sobre un solo tema".

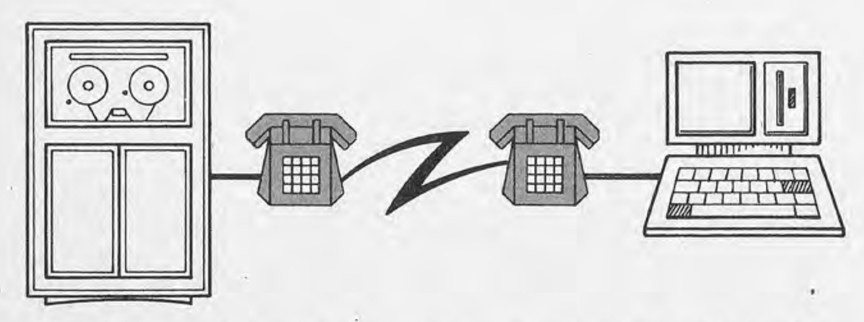


Figura 2.—De los métodos antiguos de manejo de la información a los que se utilizan en la actualidad hay una gran diferencia. La comunicación de ordenadores situados en distintos lugares ha permitido un gran avance.

Así, pues, podríamos disponer de un ordenador que sólo supiera sobre los partidos que se juegan en la Copa del Rey, y nada más que sobre eso. Si al ordenador le preguntáramos quién va el primero en la Liga, él no lo sabría, puesto que no estudia el tema "partidos de Liga".

Este tipo de tareas se realizó rápidamente, puesto que lo único que se necesitaba era memoria suficiente en el ordenador para almacenar toda la información (sobre los partidos de Copa jugados, por ejemplo). Pero alguien pensó si no sería posible que el ordenador fuera capaz de suministrar información más inteligente sobre el tema tratado: Se intentaría que el ordenador diera su opinión sobre quién va a ganar la Copa este año. Para conseguir esto hace falta que la máquina trabaje de forma inteligente con los datos que posee (que son los resultados de los partidos de Copa) para obtener una información que no poseía previamente.

Pues bien, al intento de crear un sistema basado en ordenadores que se ocupe de un determinado tema y que sea "medianamente inteligente" se le ha llamado sistema experto sobre ese tema.

Lo de "experto" puede referirse a que el ordenador se convierte en un experto sobre el tema tratado debido a la gran cantidad de información sobre esa materia que posee.

Qué es un sistema experto

Un sistema experto se puede definir, más concretamente, como una estructura de programación capaz de almacenar y utilizar, con menos restricciones de las presentes en un programa clásico, algún tipo de conocimiento sobre una determinada área.

Expliquemos un poco esta definición. Como cualquier otro paquete de trabajo sobre un ordenador, el sistema experto es una estructura de programación, es decir, es un programa. Sin embargo, no es un programa de ordenador normal como el que hacemos en el ordenador que tenemos en casa, puesto que su estructura es diferente: no tiene las mismas partes que tienen los programas clásicos.

Esta estructura de programación es capaz, básicamente, de dos cosas:

- puede almacenar información, como si fuese una base de datos, sobre el tema considerado;
- tiene la facultad de utilizar esa información para obtener unos resultados que no existían previamente en el ordenador. Esto lo realiza mediante una técnica que es la que real-

mente diferencia a los sistemas expertos de los programas clásicos.

Otras características específicas de los sistemas expertos se encuentran, además de las ya explicadas de uso general (almacenar y utilizar), en la capacidad de aprendizaje y el poder de inferencia.

La capacidad de aprendizaje de un sistema experto, aunque parece una cualidad futurista y extravagante, resulta fácil de implantar; se convierte en una herramienta muy útil a la hora de que el sistema mejore respecto del momento de su creación. En cualquier caso, la capacidad de aprendizaje es limitada y, además, debe estar supeditada al control del ingeniero que gestiona el sistema experto.

Es importante diferenciar al usuario del "ingeniero del conocimiento". El usuario es la persona que utiliza el sistema, como si fuese el dueño de una casa. El ingeniero del conocimiento se ocupa de que el sistema funcione como quiere el usuario (sería el arquitecto, el fontanero, el carpintero, el albañil... todos lo que crean la casa y la arreglan para que el dueño se sienta cómodo).



Figura 3.—Capacidades de los sistemas expertos.

De momento, y quizá por muchos años, los ordenadores no pueden sacar información *completamente* nueva respecto de la que poseen almacenada. Si a nuestro famoso ordenador, el que almacena todos los partidos que se han jugado de la Copa del Rey, le preguntamos quién cree que va a ganar la Copa del año que viene, podrá contestar cualquier equipo que ha jugado la competición, pero no podrá responder con un equipo que no existe. Esto es muy importante. Así, pues, el aprendizaje será, precisamente, lo que permita al ordenador realizar las siguientes tareas:

- completar la información almacenada;
- utilizar nueva información que necesite para procesos de inferencia.

La nueva información que aprende el sistema puede venir por dos caminos diferentes:

- Del propio usuario.
  En algunos procesos el ordenador puede ser consciente de qué nueva información puede cambiar el resultado. El ordenador "copero" podría preguntar al usuario, antes de dar alguna solución: ¿Habrá algún equipo nuevo en Primera División el año que viene? Así, en el caso de que exista alguno, éste será introducido por el propio usuario y el sistema lo incorporará a sus conocimientos y lo tendrá en cuenta como posible solución.
- De un proceso interno del ordenador.
  En este caso no se consigue información completamente nueva, sino resultante de la interrelación de información previamente acumulada.

Para ofrecer un ejemplo de este segundo método vamos a crear otro ordenador que se ocupe de resolver problemas médicos.

El usuario, en este caso un médico, le introduce al sistema los síntomas del paciente y el propio sistema expresa su opinión indicando una enfermedad. Este "médico electrónico" podría aprender de la siguiente manera: Supongamos que la información que posee el sistema sea

El frío $\xrightarrow{\text{produce}}$ catarros

Los catarros mal curados $\xrightarrow{\text{producen}}$ bronquitis

Si un paciente dice que hace una semana pasó mucho frío y que posteriormente se puso muy malo, el ordenador puede resol-

vere que el paciente posee bronquitis. Esta información, que ha inducido el propio ordenador, podría aprenderla si creara un nuevo hecho de información que fuera:

El frío $\xrightarrow{\text{produce}}$ bronquitis

Este "médico electrónico" nos sirve además para explicar brevemente la segunda gran cualidad específica de los sistemas expertos: el poder de inducción. Podemos apreciar cómo a partir de una información que ha introducido el usuario (hace una semana que pasó mucho frío) y de unas relaciones entre hechos de información que posee, el ordenador consigue una solución, combinación de todo. A la acción de combinar la información para conseguir nuevos hechos se le llama inducción.

Estructura de un sistema experto

Una vez explicada la teoría de los sistemas expertos y descritas las posibilidades y características más importantes, queda por definir cómo está formado un sistema experto. Conocer la estructura general de un sistema de este tipo permite clarificar la comprensión de su funcionamiento.

El sistema experto está formado básicamente por dos elementos:

- una base de conocimientos,
- un motor inferencial.

Analicemos cada uno de ellos:

- Base de conocimientos.
  En ella se almacena la información referente al tema tratado en *cada caso* por el sistema. La pregunta podría ser: ¿Por qué se llama base de conocimientos y no base de datos, si al fin y al cabo lo que se almacena en dicha base es información?
  Respecto a este tema todavía no se han aclarado los científicos. Incluso no se han puesto de acuerdo aún respecto a los elementos simples que engloba la propia base de conocimientos. Para profundizar en este problema hay que conocer los niveles del conocimiento diferenciados en inteligencia artificial, como haremos posteriormente.
- Motor inferencial.
  Se ocupa de gestionar la base de conocimientos de modo que ante preguntas del usuario el sistema ofrezca respuestas aceptables. Es decir, se ocupa de tomar las reglas y realizar inferencias para que tengamos una solución. Más adelante se explicarán estas funciones con más detalle.

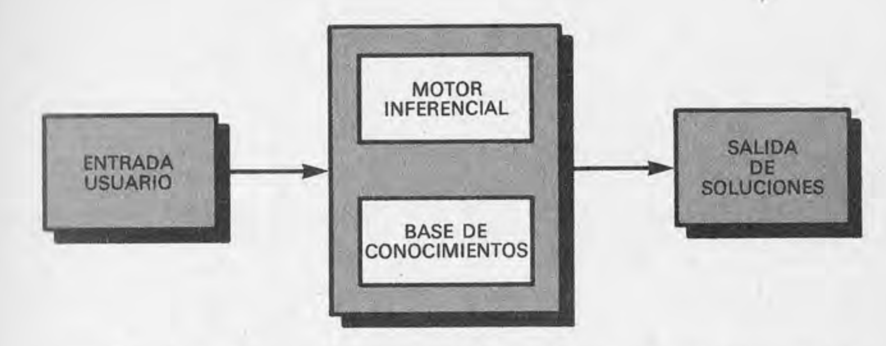


Figura 4.—El esquema general de un sistema experto presenta un diagrama de bloques bastante simple.

Niveles de conocimiento

Existen tres niveles de conocimiento que estructuran la información que maneja un sistema experto.

- **HECHOS BASICOS DE INFORMACION**

Es el nivel más bajo dentro de la jerarquía de conocimientos. Representan simplemente hechos básicos, es decir, los elementos lingüísticos más simples que poseen sentido por ellos mismos. Son la base de la información.

Ejemplos de hechos básicos son:

"Nuria está como una foca"
"Nacho tiene gafas"
"Marcos es diplomático"

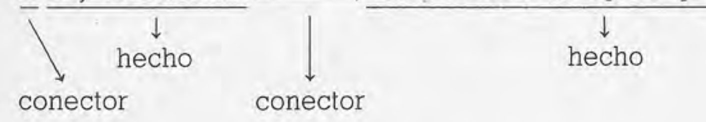
- **REGLAS DE CONOCIMIENTO**

Son bloques lingüísticos que representan un determinado tipo de información en función de la forma en que se construyan. Están formados por hechos básicos, aunque puede ocurrir (de hecho ocurre en algunos sistemas expertos) que las reglas de conocimiento pueden por ellas mismas representar la unidad mínima de conocimiento con la que trabaje un determinado sistema

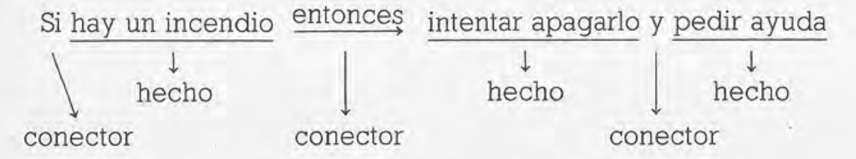
experto. Una regla de conocimiento puede representar lo que hay que hacer en caso de incendio; y solamente representará eso. En el caso de que ocurriera una inundación no podríamos aplicar esta regla.

Para crearla habrá que utilizar como mínimo dos hechos básicos: el primero representará qué hacer realmente en caso de incendio; el segundo servirá para tener claro que la regla creada sólo se debe utilizar cuando hay un incendio y no cuando hay una tormenta, por ejemplo.

Así podríamos crear la siguiente regla:



Aunque podríamos construir otra regla como:



donde la propia regla está formada por más de dos hechos.

Vemos como mediante la información de las reglas de conocimiento se constituye el conocimiento del sistema sobre un tema escogido (en el caso anterior, sería cómo tratar un incendio).

Aunque se explicará posteriormente el significado de los conectores, vemos ya que su función es unir diferentes hechos para crear una estructura (la regla de conocimiento) que posea sentido en su conjunto y que además éste debe ser el que pretendemos que tenga.

Se puede ver que la estructura básica de las reglas de conocimiento es:

Si [hechos-1] entonces [hechos-2]

donde [hechos-1] es el primer conjunto de hechos, que recibe el nombre de antecedente, y [hechos-2] es el segundo, que se le llama consecuente.

- **REGLAS DE CONTROL**

Representan el último nivel de información en la estructura del conocimiento aquí tratada. Una vez constituidas las reglas de

conocimiento sobre un tema, se almacenan todas ellas en la memoria del ordenador. Sin embargo, ante una pregunta del usuario habrá que utilizar, para responderle, sólo unas cuantas de las reglas que se poseen, no todas.

¿Qué reglas deben de utilizarse para contestar al usuario? El sistema necesita un instrumento capaz de elegir en cada caso la regla que corresponda.

Siguiendo con el ejemplo del incendio, supongamos que en el ordenador tenemos dos reglas de conocimiento:

Regla A: "Si hay un incendio $\xrightarrow{\text{entonces}}$ corre muy deprisa"

Regla B: "Si hay una inundación $\xrightarrow{\text{entonces}}$ aprende rápidamente a nadar"

En cada caso el ordenador debería analizar las reglas y escoger la que realmente tiene que aplicar. Sería muy problemático que alguien en un incendio intentara aprender a nadar (aunque probablemente no se quemaría).

La elección de las reglas adecuadas para cada caso se realiza en el ordenador por medio de otras reglas, llamadas reglas de control.

Las reglas de control, como su nombre indica, controlan la elección de las reglas de conocimiento. Por ejemplo, una regla de control podría ser:
"Cuando ocurra un hecho «A» entonces aplicar todas las reglas en que aparezca como antecedente el propio hecho «A».

Así, en el caso de que ocurriera un incendio se aplicarían todas las reglas donde apareciese:
"Si hay un incendio..."

La estructura de conocimientos utilizada se engloba en la forma que ilustre la figura 5.

Asignación de niveles

Una vez definidos los niveles de conocimiento, el siguiente paso es asignarles un lugar en la estructura del Sistema Experto.

Aquí empiezan las divergencias de opinión entre los científicos.

Para algunos dicha asignación es:

Hechos básicos $\xrightarrow{\text{forman la}}$ base de datos asociada al sistema

Reglas de conocimiento $\xrightarrow{\text{forman la}}$ base de conocimientos

Reglas de control $\xrightarrow{\text{forman el}}$ motor de inferencia

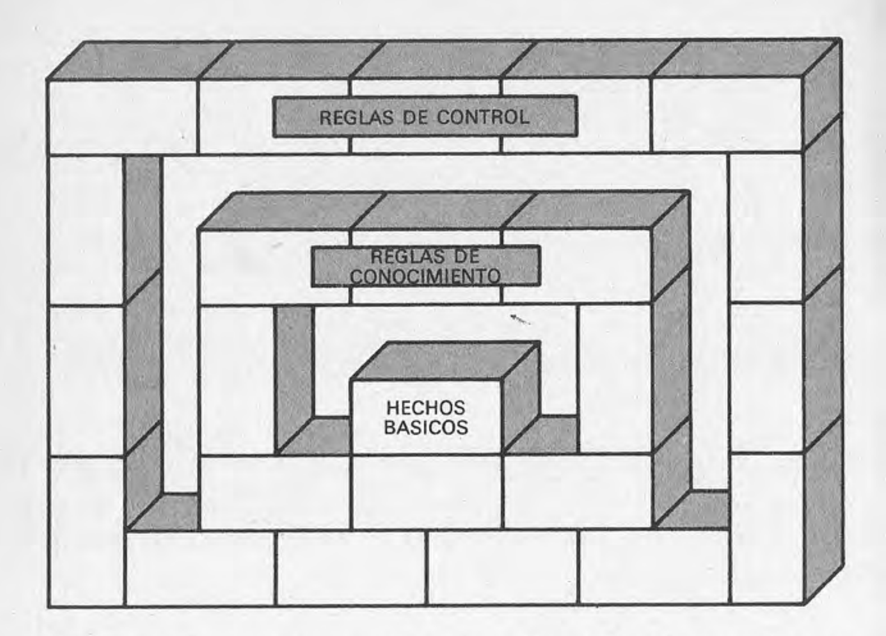


Figura 5.—La estructura que poseen los niveles de conocimientos de un sistema experto tiene una forma envolvente.

Es decir, crean una base de conocimientos formada por reglas y, además, el sistema posee una base de datos asociada.

Para otro grupo de científicos la asignación se realiza de la siguiente manera:

Hechos básicos
Reglas de conocimiento } $\xrightarrow{\text{forman la}}$ base de conocimientos

Reglas de control $\xrightarrow{\text{forman el}}$ motor de inferencia

En esta asignación la base de conocimientos está formada por los hechos básicos y por las reglas de control.

Para manejar correctamente la inferencia existen gran cantidad de herramientas lógicas y matemáticas estudiadas en este siglo. Entre ellas se encuentran la deducción, la inducción, etc. Todos estos métodos matemáticos pueden ser implantados en los sistemas para que trabajen con las reglas y produzcan soluciones.

Una vez estudiadas la estructura y cualidades más importantes de los sistemas expertos podemos realizar un gráfico que aglu-

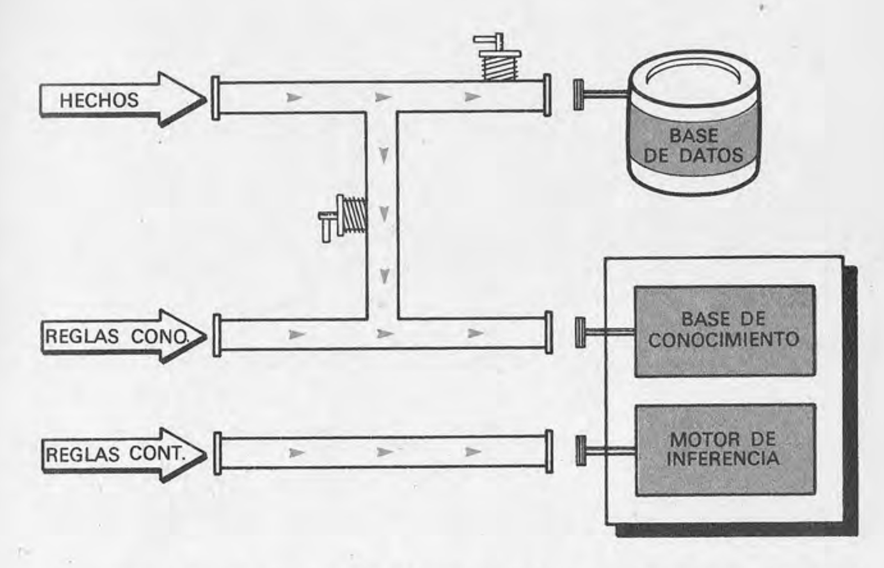


Figura 6.—La asignación de los niveles de conocimiento a cada elemento del sistema experto se puede representar mediante una canalización de tuberías.

tine perfectamente el trabajo del sistema, tal como el mostrado en la figura 7.

Si nos hacemos la famosa pregunta: ¿por qué a la base de conocimientos no se le llama base de datos? comprenderíamos ahora que en los sistemas expertos la base de conocimientos significa algo más que información almacenada, puesto que incluye ya una cierta "inteligencia" en cómo y cuándo utilizar la información. Por eso se llama precisamente *Base de Conocimientos.*

Diferencias estructurales con la programación clásica

Los sistemas expertos representan un paso adelante en la ciencia de la programación por su gran diferencia con los programas clásicos.

En la programación normal para resolver un problema se crea un programa. En este programa, que está constituido por líneas en BASIC, FORTRAN u otro lenguaje, se encuentran formando una unidad tanto la información sobre el método para resolver el problema como los propios datos particulares del problema.

Sin embargo, en los programas de inteligencia artificial (y más concretamente en los que constituyen los sistemas expertos) se

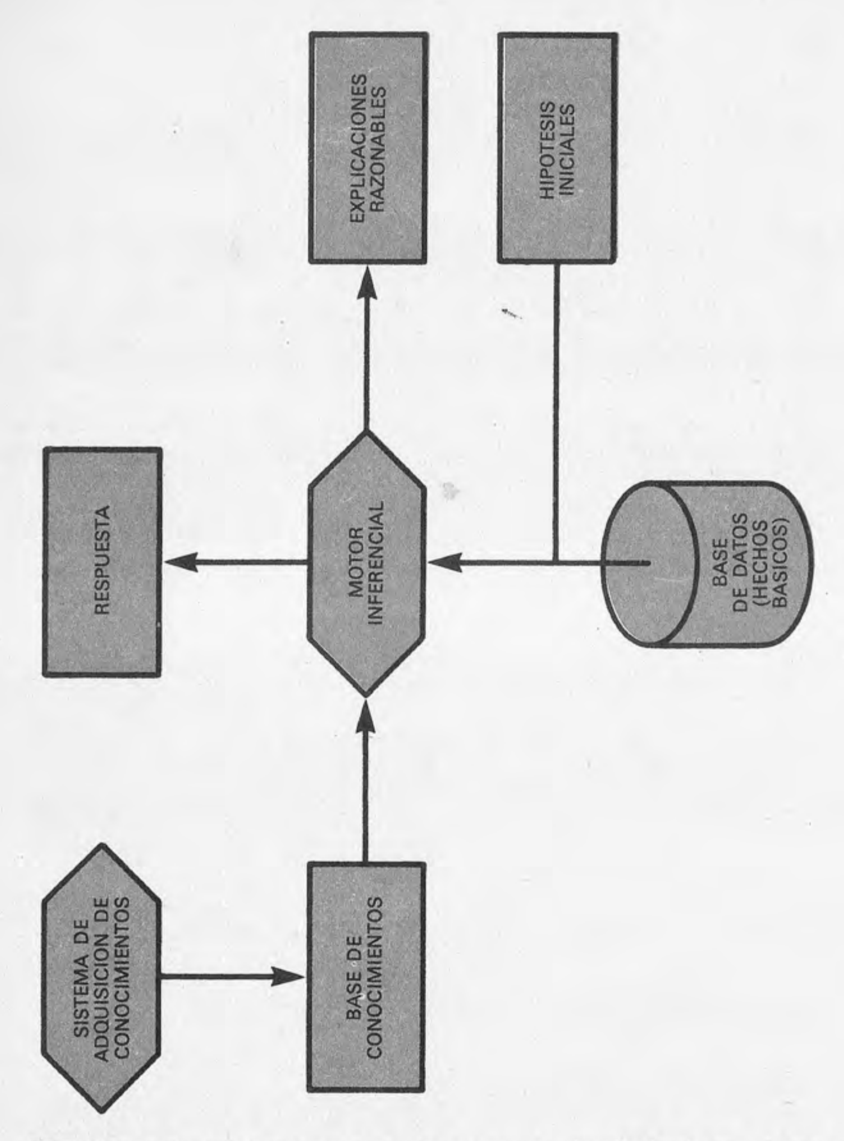


Figura 7.—La figura expresa la estructura detallada de un sistema experto en función de las facilidades que presenta en su funcionamiento.

consigue diferenciar el método de resolución de problemas (una especie de "solucionador de problemas" pero sin problema que resolver) de la información característica del problema. Esto es

muy importante, puesto que facilita enormemente la evolución del sistema al ser modulada su estructura.

Así pues, el motor de inferencia se construye como un ente aislado e independiente de la base de conocimientos. Esto permite crearlo a la vez que otras partes del sistema y así se agiliza el proceso necesario.

Sin embargo, hay que reconocer que no existe un 100% de independencia entre la base de conocimientos y el motor de inferencia, ya que, como ocurre con todo en esta vida, unos motores inferenciales funcionan mejor con un determinado tipo de base de conocimiento que con otro y, por tanto, se crean unas pequeñas diferencias entre los motores en función de la forma que toman el conocimiento sobre el tema.

Ya tenemos definidas las diferencias de estructura entre los programas clásicos y los sistemas expertos. Sin embargo, todavía no está muy clara la gran ventaja de estos sistemas. Pueden pensar ustedes que mucho definir y estructurar pero poco diferenciar realmente su funcionamiento; por lo visto hasta ahora se podría haber hecho un programa normal estudiando por un lado la

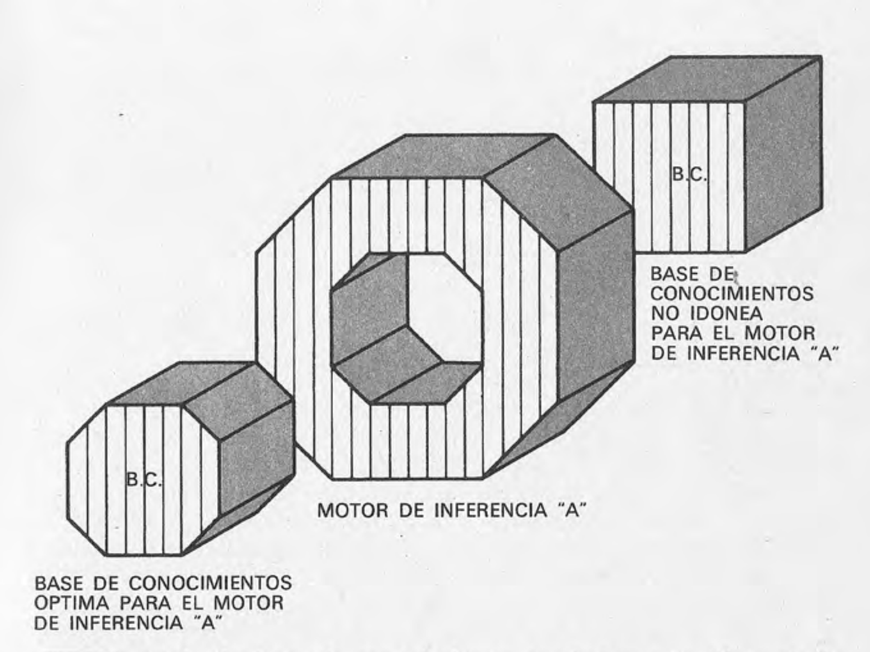


Figura 8.—La base de conocimientos es el elemento diferenciador de los múltiples sistemas expertos. No todos se adaptan igual a un mismo motor inferencial.

mecánica de resolución y almacenando por otro los datos del programa. Sin embargo, "sí" existen grandes diferencias de funcionamiento.

Estructura Sol

Supongamos que queremos realizar un sistema experto en diagnósticos médicos; volvemos a trabajar con nuestro Médico Electrónico. El sistema experto que vamos a realizar, como somos muy "chulos", podrá resolver cualquier caso sobre cualquier tipo de enfermedad (¡imagínense lo que pedimos!); aunque no se ha conseguido en realidad, nosotros ya lo tenemos.

Para enseñar al sistema la primera enfermedad que debe conocer tendríamos que recoger todos los síntomas que conocemos sobre ella para luego introducirlos en el ordenador. Tomemos, por ejemplo, la gripe:

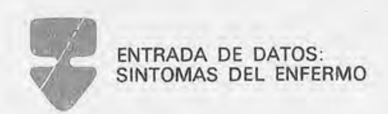
- fiebre no muy alta;
- malestar general;
- dolores en todo el cuerpo;
- tos, etc.

Pero ¿quién nos impide realizar un programa clásico que compruebe si tenemos la gripe? Podría tratarse de un programa que funcionara de forma que: Si entre los datos de entrada que dé el usuario se encuentra algún síntoma que defina la gripe entonces diría simplemente que tenemos la gripe. En caso contrario concluiría que no tenemos la gripe.

Esta estructura (Fig. 9) es la normal en un programa clásico. ¿Qué ocurre si además de analizar si el paciente sufre la gripe, el sistema estudia la posibilidad de que tenga artritis? De nuevo lo primero es recopilar los síntomas de la artritis (dolor de articulaciones y huesos, pérdida de movilidad etc.); entonces se podría crear otro "programa" para reconocer si el paciente posee artritis, que se podría juntar con el que teníamos. Así podríamos saber si el paciente posee gripe o artritis.

Pero existe un problema: el programa global debe analizar *en primer lugar una de las dos enfermedades*. Es decir, deberá ver primero si el paciente tiene la gripe y luego analizar la posibilidad de que sufra artritis (Fig. 10), o al revés.

Así se podría continuar con todas las enfermedades que conocemos. Esto da la impresión de que cualquier programa realizado de forma clásica podría conseguir los mismos resultados finales que uno con técnicas de inteligencia artificial.



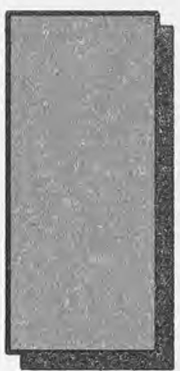




Figura 9.—Estructura de un programa clásico. El bloque básico de inferencia corresponde a un programa simple que realiza una única función.

Sin embargo, la solución final lograda mediante la programación clásica resultaría muy poco compacta y su funcionamiento sería muy lento al tener que recorrer de una forma predefinida todo el espectro de enfermedades que el programa global tuviera almacenado.

Analicemos la forma que va tomando el bloque global de inferencia. Para el caso de la doble posibilidad, anteriormente analizado, se tiene el bloque de la figura 11.

En el caso de que se tuviera un programa con más enfermedades, es decir con más bloques de inferencia, se conseguiría una zona de inferencia como la de la figura 12.

Los sistemas expertos son mucho más eficaces que los programas clásicos porque pueden activar el bloque de inferencia que se necesita en cada caso sin necesidad de tener que analizar todos los precedentes.

Es decir: si el paciente tiene hepatitis, el sistema experto analizará los datos de entrada que se le han introducido y deducirá que tiene hepatitis sin deducir previamente que no tiene gripe y que no tiene artritis. En este punto radica la gran ventaja de los

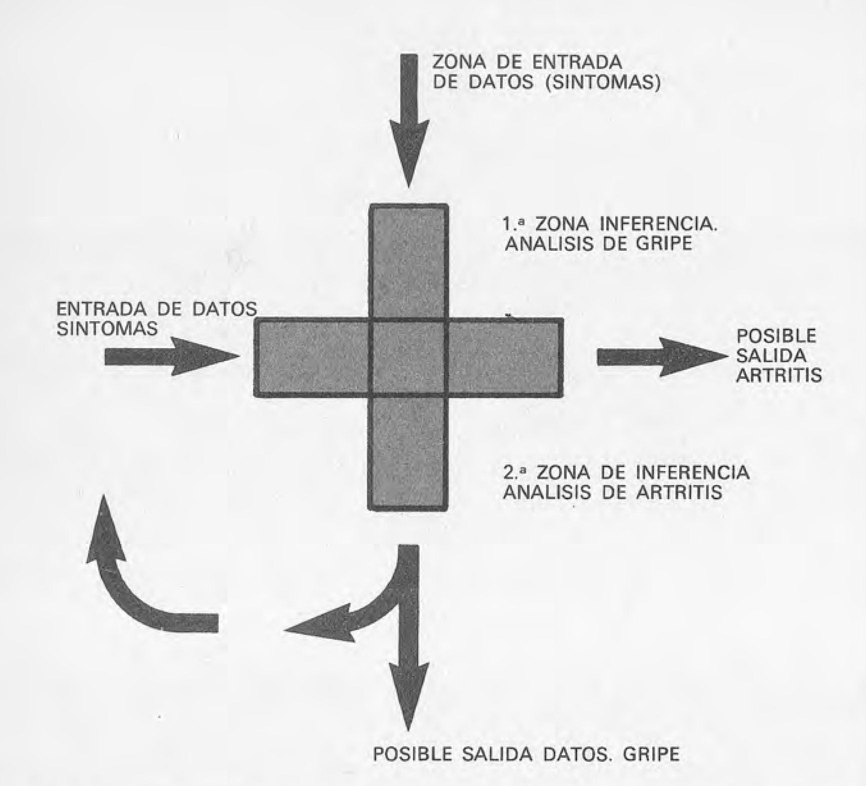


Figura 10.—El bloque de inferencia corresponde a un equivalente de dos programas clásicos funcionando uno después de otro. Primero se decide una cosa y luego otra.

sistemas expertos. De una base de conocimientos formada por mucha información sólo se activa la que corresponde a cada caso en función de la entrada de dagos. Esto es lo que se conoce como estructura sol (Fig. 13).

A partir de la estructura sol del sistema experto se puede comprender perfectamente el papel fundamental del motor de inferencia. Ante cada problema planteado por el usuario y teniendo en cuenta la estructura autónoma del sistema, se pueden activar gran cantidad de bloques de inferencia para resolverlo.

Estos bloques no tienen por qué ser los anteriormente definidos, sino que simplemente son formados por reglas independientes unas de otras, pero encadenadas para resolver el problema específico que se plantea en cada caso.

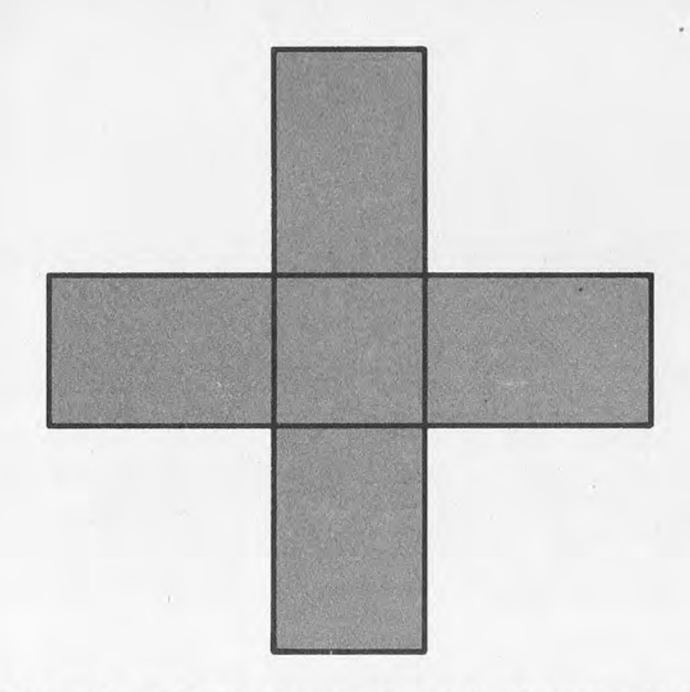


Figura 11.—La figura representa la zona de inferencia utilizada y su representación particular en un programa clásico con dos posibilidades.

La base de conocimientos se presenta como un laberinto (figura 14). En función de la puerta por la que se entre y en función de las normas básicas que utilicemos para atravesarla saldremos a un lugar o a otro, obteniendo resultados distintos en cada caso.

Sin embargo, entre todos los caminos posibles para resolver un problema hay uno que resultará el mejor. Para conseguirlo habrá que utilizar las reglas óptimas de la base de conocimientos: el motor de inferencia deberá elegirlas. Esta tarea resulta fundamental y es precisamente en este punto del problema donde se plantean soluciones innovadoras.

Para elegir las reglas óptimas se pueden aplicar muchos criterios: simplemente mecánicos, es decir, sin que exista ninguna "chispa de inteligencia" o bien más heurísticos, basados en la experiencia. La palabra anteriormente citada, heurística, representa el eje sobre el que gira la inteligencia artificial y se estudiará con más profundidad posteriormente.

Existe en la actualidad una tendencia a creer que todo problema informático *debe* resolverse mediante la creación de un sis-

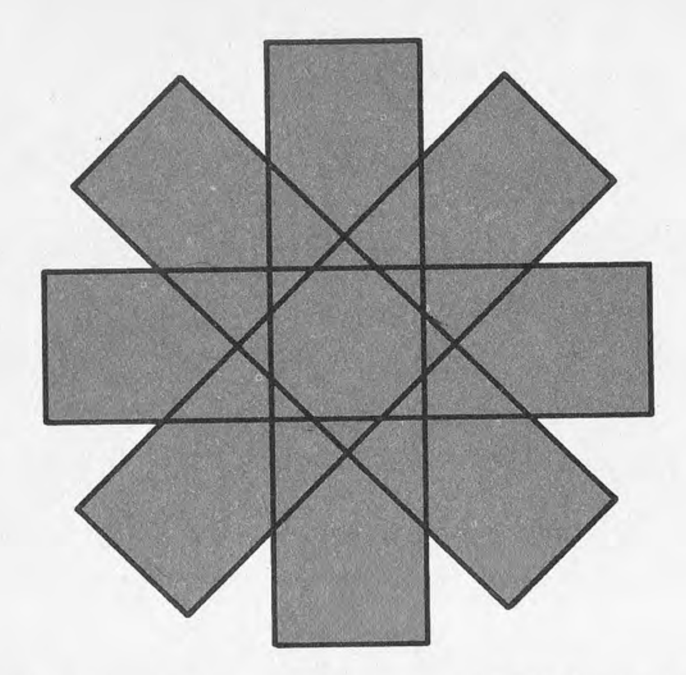


Figura 12.—Según van aumentando las posibilidades en un programa tradicional, las zonas de inferencia se multiplican.

tema experto específico y esto es falso. Es cierto que muchos problemas podrían tener una solución desde el ángulo de la inteligencia artificial, pero ésta puede llegar a ser en ocasiones mucho menos eficiente que un programa clásico. Esto es así porque existe un determinado tipo de problemas que pertenecen al entorno de la inteligencia artificial y otro que no encaja tan bien. Por tanto, antes de realizar un proyecto informático conviene hacer un estudio de viabilidad y elección de las herramientas adecuadas para su ejecución. No cuesta demasiado y puede ahorrar mucho tiempo y dinero.

Particularidades de los Sistemas Expertos

Debido a la gran cantidad de posibilidades a la hora de crear el "camino de inferencia" para la resolución de un problema puede ocurrir (y de hecho se espera que ocurra) que ante los problemas planteados se consigan soluciones no programadas.

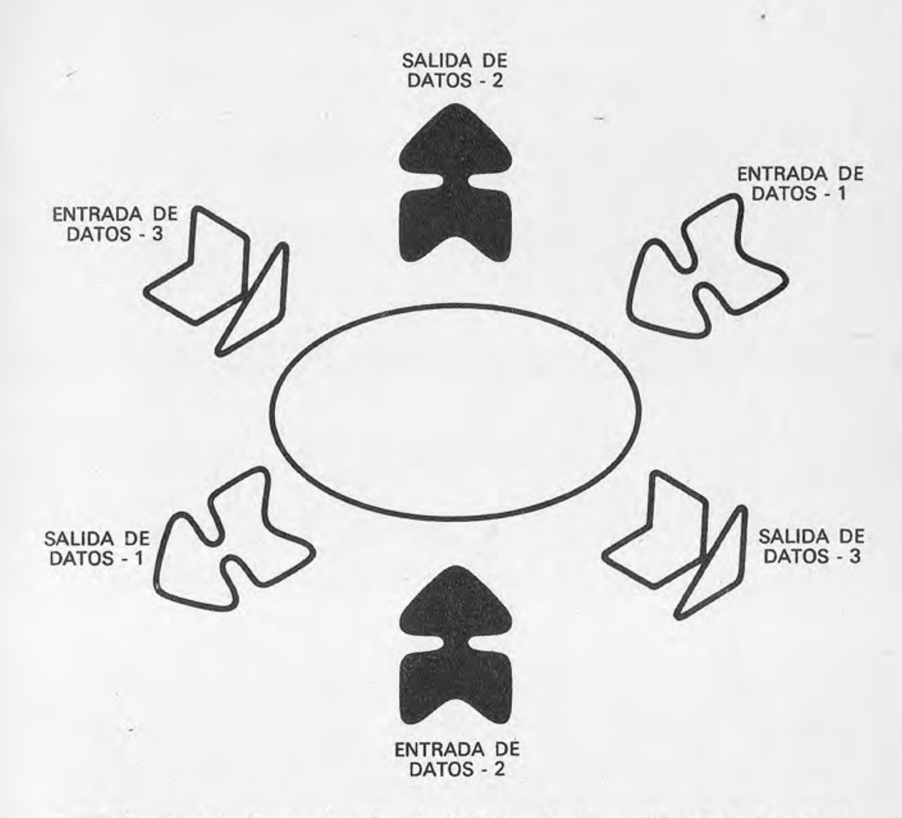


Figura 13.—La evolución de la base de conocimientos vista como conjunto de bloques de inferencia da como resultado una estructura en sol del sistema experto.

Esto es precisamente lo que se pretende, que el sistema posea "movilidad", que sea algo más que una simple base de datos de soluciones.

Gracias a su estructura el sistema es capaz de explicar, paso por paso, su evolución desde las condiciones iniciales hasta la solución final y, además, describir por qué ha tomado cada decisión a la hora de las sucesivas opciones dentro del "camino de inferencia".

Como la base de conocimientos se encuentra separada del resto del sistema éste puede renovarse, actualizarse y mejorarse en su información sin necesidad de variar su estructura operativa. Esto resulta muy importante, puesto que puede utilizarse incluso para resolver tareas completamente distintas (dentro de la moda-

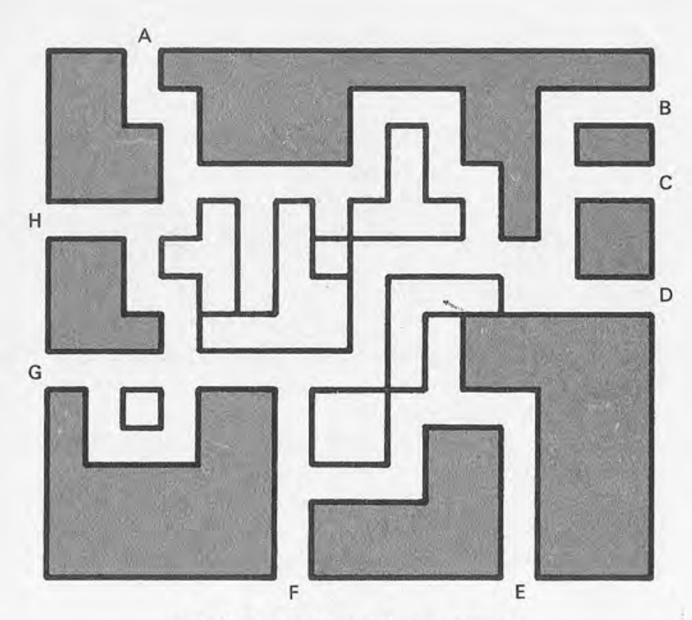


Figura 14.—Una cualidad importante de la base de conocimientos es que posee muchas "entradas" y "salidas". En cada caso la salida final dependerá de la entrada elegida y de las reglas usadas. En este sentido es comparable a un laberinto.

lidad de problemas aptos para ser resueltos con la inteligencia artificial).

Areas de aplicación de los Sistemas Expertos

Hemos hablado de problemas que se prestan perfectamente a su resolución mediante técnicas de inteligencia artificial, de motores de inferencia que funcionan mejor con un tipo de bases de conocimiento que con otras y de un largo número de diferencias sustancialles.

Esto nos hace pensar que existirán diversas zonas de aplicación para sistemas expertos en función del tipo de diseño a que hayan sido sometidos.

Cada una de estas zonas es cubierta por un tipo de sistema

experto que posee las condiciones óptimas para cubrirla. Se consideran dos tipos básicos de sistemas expertos:

- **SISTEMAS EXPERTOS GENERATIVOS**

Como su nombre indica realizan tareas de tipo "creativo" (en la concepción más limitada de la palabra). Se les intenta introducir los máximos niveles de inteligencia para que dependan lo menos posible de las directrices fundamentales de cada caso introducidas por el "hombre". La información que utilizan está compuesta por la que le introduce el usuario para cada problema y por la almacenada en su propia base de conocimientos.

Su principal función es el diseño (por ejemplo, industrial, de piezas mecánicas, etc.), la inducción de problemas matemáticos, etc.

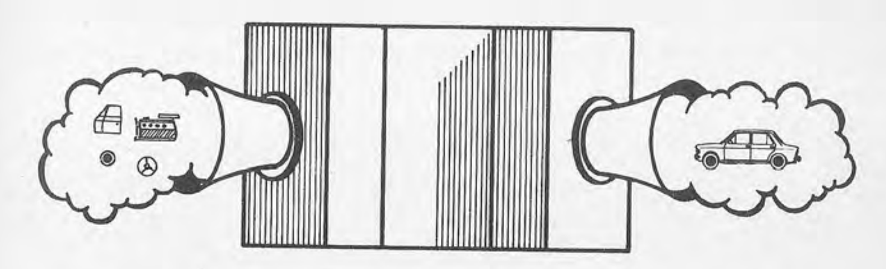


Figura 15.—El sistema experto generativo "crea" soluciones a partir de los datos de entrada.

- **SISTEMAS EXPERTOS OPERATIVOS O INTERPRETATIVOS**

Su forma de trabajo es menos original que la de los sistemas generativos. La información que manejan proviene básicamente de la que el ordenador posee almacenada; el peso específico de la información que suministra el usuario es menor para valorar la actuación del sistema. Son sistemas de tipo consultivo.

Manejan problemas que no se presentan en los otros sistemas, como son el manejo de gran cantidad de información, la correcta gestión de ésta y problemas de precisión y certeza en la información que el sistema presenta como solución al problema planteado.

Los principales campos de operación para este tipo de sistemas son el mantenimiento de máquinas, la consulta médica especializada, etc.

Por ejemplo, nuestro sistema "Médico Electrónico" formaría parte de este grupo, puesto que lo que realiza es un diagnóstico médico del paciente.

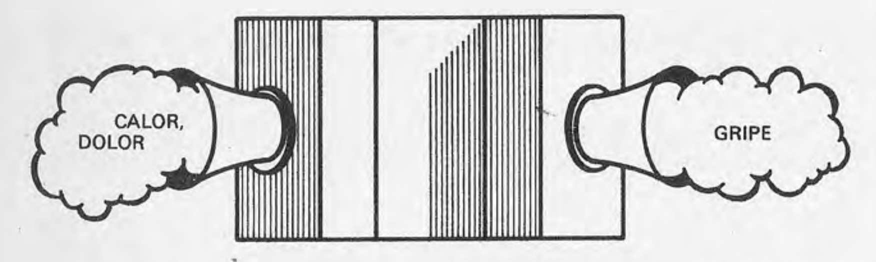


Figura 16.—El sistema experto interpretativo "interpreta" los datos de entrada hasta lograr dar con una solución convincente.

Ventajas de los sistemas expertos

La principal ventaja de los sistemas expertos es que, cuando realmente sirven para lo que se les construye, optimizan cualquier tarea acometida respecto de un programa convencional en un tiempo semejante.

Clasificando estas ventajas se tendría:

AYUDAR

El sistema experto puede responder a preguntas que le formulen de una forma directa cuando posee las soluciones almacenadas en su memoria. Es decir, puede servir como una ayuda tipo agenda.

Su memoria se utiliza como la del usuario, puesto que los datos que éste pide se encuentran almacenados en el ordenador.

Por ejemplo, creemos un tercer sistema experto, al que llamaremos "Cuida-coches electrónico". Se ocupará de mantener nuestro coche en perfecto estado para la conducción mediante la información de "cómo se encuentra el coche en cada instante", mira los kilómetros, el aceite, el agua, las ruedas, etc., y deduce si al coche hay que hacerle alguna reparación, cambiarle alguna pieza o funciona bien. Además, como nuestro "cuida-coches electrónico" es un sistema experto estupendo, podemos preguntarle cualquier cosa sobre el funcionamiento del vehículo.

Una de las reglas que tendría el sistema sería:

REGLA 1:
SI EL NIVEL DE ACEITE DEL COCHE ES BAJO $\xrightarrow{\text{entonces}}$ EL MOTOR SE GRIPA

La capacidad de ayuda del sistema experto se pondrá de manifiesto cuando le preguntemos:

¿Qué ocurre si el nivel de aceite es bajo?

puesto que contestará:

"El motor se gripa"

No ha hecho ningún tipo de inferencia, puesto que la información se encuentra "tal cual" en el ordenador.

INFERIR

Representa un paso más evolucionado. El sistema es capaz de responder a preguntas con soluciones que no se encuentran directamente almacenadas en la memoria del ordenador. Esto se consigue mediante un proceso de inferencia, de modo que a partir de las premisas iniciales se consiguen hechos que solucionan el problema planteado.

Supongamos que nuestro sistema posee, además de la regla anteriormente citada, otra regla que dice:

REGLA 2:
SI EL MOTOR SE GRIPA $\xrightarrow{\text{entonces}}$ EL COCHE SE ROMPE

Un usuario le podría preguntar:

¿Qué pasaría si el nivel de aceite es muy bajo?

y el ordenador contestaría:

"El coche se rompería"

Podemos ver cómo la información resultante de la utilización del sistema ("si el nivel de aceite es muy bajo puede ocurrir que el coche se rompa") no se encontraba previamente almacenada en el ordenador, sino que ha sido inducida en su funcionamiento normal.

JUSTIFICAR

El poder de justificación del sistema se entiende como la facilidad de exponer todos los pasos del razonamiento realizado por el ordenador sobre la base de conocimientos incorporados y la modificación de hechos básicos para tener una idea exacta del por qué de la solución escogida.

En nuestro sistema, para continuar con el ejemplo, ante la respuesta que nos ha proporcionado el sistema:

"Ante un nivel de aceite muy bajo ocurriría que el coche se rompería"

le podríamos pedir que la justificara

"Si el nivel de aceite es bajo el motor se griparía y si se gripara el motor entonces el coche se rompería"

Con esto la conclusión estaría completamente justificada.

APRENDER

Esto es un aspecto poco explicado. La gente que desconoce el tema siente un cierto temor (no falto de fundamento "en el fondo") ante la posibilidad de que un sistema artificial sea capaz de aprender "algo". Sin embargo, la capacidad de aprendizaje de estos sistemas expertos es ciertamente limitada y, además, muy artificial.

Un aspecto que no se ha comentado todavía de los sistemas expertos es el del aprendizaje inicial que tiene el "programa" antes de que su funcionamiento pueda ser considerado como óptimo. Este aprendizaje inicial, fundamentalmente realizado por el ingeniero del conocimiento, consiste en enseñar al sistema lo necesario para que éste pueda trabajar con conocimientos sobre el tema tratado. En la realidad lo que se hace es llenar la base de conocimientos con la información básica sobre el tema. Con esto se evitará que el sistema experto se comporte como un verdadero "inutil" antes de haber aprendido lo suficiente. Este primer aprendizaje representa, por tanto, la toma de nivel.

Sin embargo, existe otro aprendizaje que sirve para mejorar ese nivel inicial y conseguir que el sistema evolucione, lo que tiene lugar durante la fase de utilización: en él cualquier inferencia puede ser almacenada, con lo que se amplía su conocimiento.

Otra forma de aprendizaje, mucho más artificial todavía, consiste en que cuando el sistema no sabe algo se lo pregunta al usuario.

Siguiendo con el ejemplo de nuestro "cuida-coches electrónico", la inferencia que se ha conseguido:

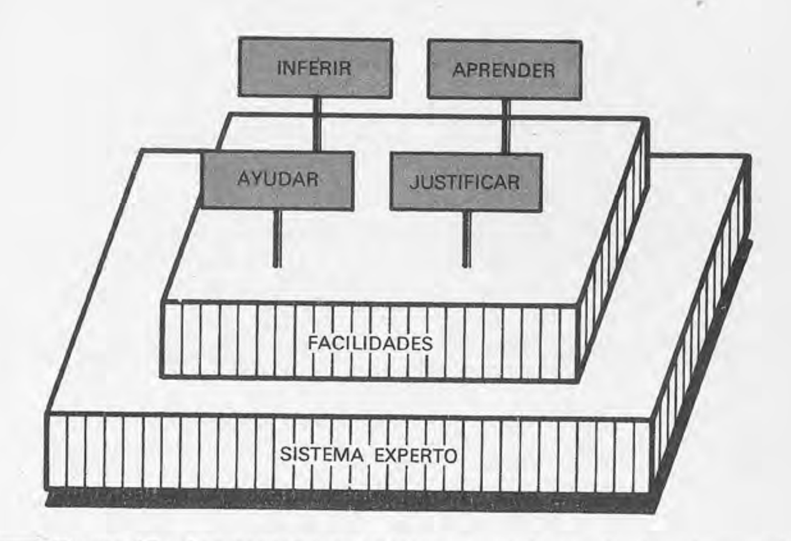


Figura 17.—Las ventajas del sistema experto son las banderas a favor de la Inteligencia Artificial.

"Si el nivel del aceite está muy bajo entonces el coche se rompe"

podría almacenarse como una nueva regla de la base de conocimientos correspondiente. El sistema habría aprendido.

Una aplicación de los S.E.: La recuperación de información

Una aplicación muy importante de los sistemas expertos se encuentra en un campo que posee un enorme futuro como foco de desarrollo: Se trata de la recuperación de información, campo que ha surgido como continuación de los intentos que se realizaron para aglutinar la información de todo tipo en bases de datos. Consiste en el estudio de los métodos presentes y futuros cuyo fin es optimizar la recogida de información, almacenada en bases de datos, con destino al usuario final.

Actualmente existen muchos problemas, no sólo con la poca capacidad de las propias bases, sino con su especificidad, su forma de almacenar la información y la manera de acceder a ella desde el exterior. Justamente para intentar mejorar todo esto se está aplicando la I.A.

Los sistemas expertos tienen mucha utilidad en este campo, puesto que pueden servir de intermediarios entre las bases de datos y el usuario final. Así, por ejemplo, hoy día es muy difícil con-

sultar una base de datos con un lenguaje "natural", es decir, un lenguaje normal (hablado o escrito), sino que hay que hacerlo mediante un lenguaje de comandos más o menos complicado.

Podría construirse un sistema experto que tuviera como finalidad el poder entender lo que el usuario le pregunte (en su propio idioma) y luego traducirlo al lenguaje de comandos que la base de datos necesita. Con esto se conseguiría facilitar las cosas a la hora de que una persona sin conocimientos de informática pudiera utilizar el ordenador como si fuera una simple máquina de escribir.

Otra tarea de este sistema experto intermediario podría consistir en llevar a cabo el proceso contrario, es decir, que ante unas soluciones (información) enviadas por la base de datos al usuario

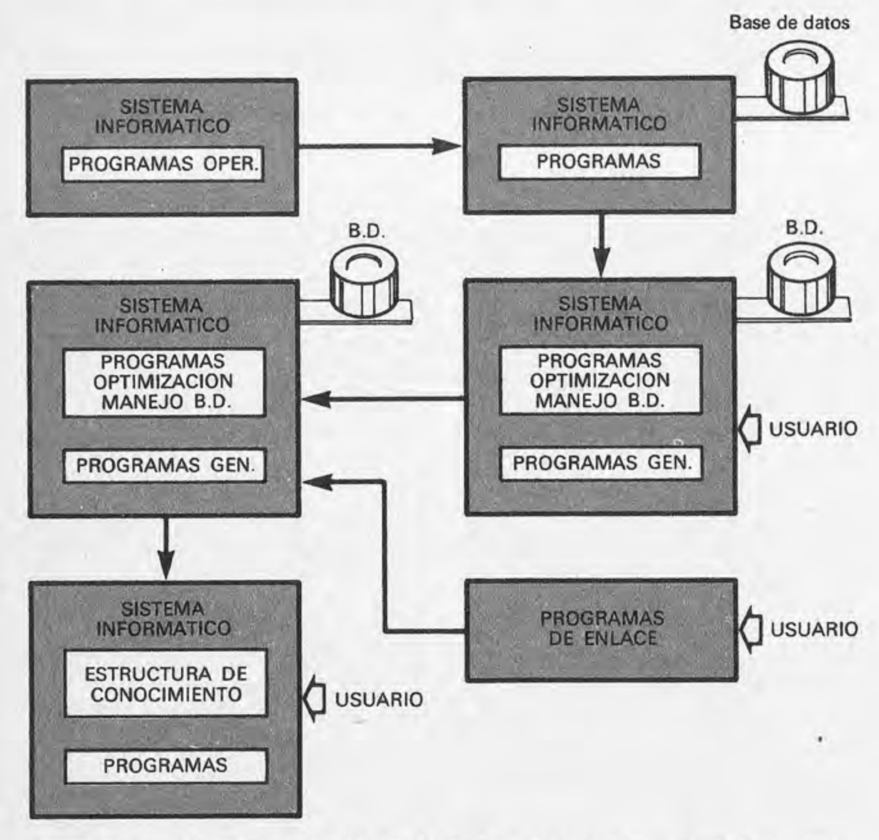


Figura 18.—La evolución final de los sistemas expertos de recuperación de información permitirá un manejo óptimo del conocimiento.

en forma de comandos se consiguiera que el sistema los transformara de forma que se pudieran leer como una novela. Con esto se evitaría de nuevo el tener que conocer aspectos internos de las bases de datos para hacer uso de ellas.

Ante un futuro que cada vez se ve más próximo, se plantea la duda de hacia dónde van a evolucionar los sistemas de recuperación de información. Parece claro que las actuales bases de datos desaparecerán, dejando paso a sistemas expertos en los cuales la información se almacene en la base de conocimientos. Con esto se dará un gran paso, no sólo en la optimización de búsqueda que se conseguiría, sino porque la información podría rentabilizarse muchos más. En una base de datos ésta debe de ser estructurada mediante procedimientos asociativos, es decir, bien mediante relación de unos hechos con otros, bien de forma jerárquica, etc. Sin embargo, la base de conocimientos permite almacenar "conocimientos" independientes unos de otros, lo cual representa una gran ventaja.

La evolución de los sistemas expertos de recuperación de información se correspondería con el gráfico de la figura 18.

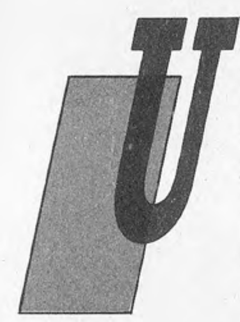
Teniendo en cuenta que en la figura está la evolución de los sistemas informáticos vista de menor a mayor complejidad y no a través del tiempo, podemos observar cómo lo fundamental del sistema se encuentra en su capacidad interna de operación mediante unos programas. En un siguiente paso, se le "cuelga" al sistema una base de datos para que su trabajo no sea puramente interno, sino que posea conexión con el mundo exterior. Lo siguiente que se intenta es, por supuesto, optimizar esa conexión para que el sistema le saque el máximo rendimiento.

Posteriormente, una vez mejorada la comunicación con el mundo exterior, se intenta conseguir optimizar la conexión del sistema con el usuario para que éste no tenga que conocer ningún tipo de lenguaje especial para comunicarse con el ordenador.

Por fin el paso del futuro representa el momento en que todos los bloques anteriormente citados se integran en una sola estructura.

CAPITULO III

REPRESENTACION DEL CONOCIMIENTO



no de los puntos débiles de toda la teoría de programación que hemos estado viendo hasta ahora estriba en la forma de representar el conocimiento. Hay que buscar una capaz de representarlo lo suficientemente bien como para que resulte efectiva y aprovechable. Parece una tontería, pero no lo es; así, supongamos que queremos representar la siguiente afirmación:

"Hoy llueve mucho, pero ayer llovió más de lo que lloverá mañana".

Desde luego, hasta a una persona le resultaría difícil *cuantificar* una información como la anterior, así que si intentamos introducirla en un ordenador los problemas serán aún mayores. De entrada parece imposible que el ordenador sea capaz de saber cuánto significa realmente la palabra "mucho". Lo que nosotros hacemos indirectamente cuando nos comentan "hoy llueve mucho" es recurrir a una información previa que tenemos almacenada de antemano y que adquirimos con la experiencia. Sabemos que cuando llueve mucho el cielo está muy cerrado y el agua que cae empapa el cuerpo muy rápidamente. Es decir, tenemos un conocimiento aproximado del agua que cae cuando llueve mucho; cuando decimos "llueve mucho" sabemos que cae más de una gota cada hora y, desde luego, menos que si nos cayera todo el mar Mediterráneo encima de golpe.

Pues lo mismo debe tener un ordenador para ser capaz de manejar este tipo de conceptos. Sin embargo, es muy difícil de conseguir, puesto que el hombre lo va aprendiendo desde su nacimiento, de tal modo que entra a formar parte de su vida.

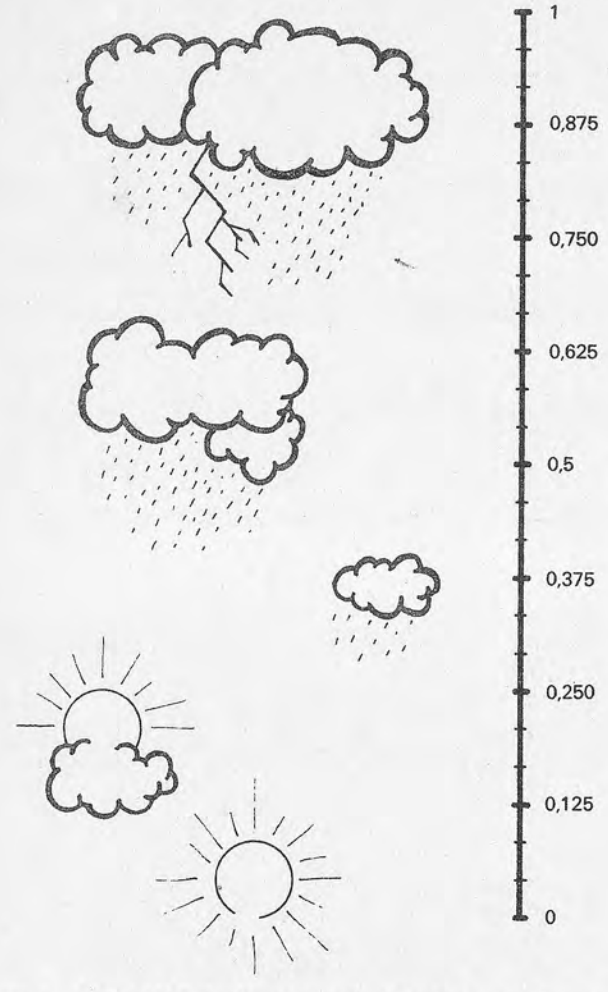


Figura 1.—El "hecho" de llover puede intentar cuantificarse matemáticamente mediante una escala graduada.

Por esto todos los métodos actuales de representación del conocimiento son aproximados y no encajan exactamente con lo que el hombre desea realmente transmitir.

Así, para representar la frase anterior podríamos introducir en el ordenador algo parecido a los siguiente:

Hoy llueve 0.6 pero ayer llovió 0.5 y mañana lloverá 0.3

Hemos creado una tabla para la lluvia donde el volumen está comprendido en un intervalo (0-1). El 0 significa que no hay un sol espléndido y el 1 que está cayendo un aguacero que produce inundaciones.

Hemos generado, pues, una relación numérica entre la información que queremos transmitir y su cuantía (Fig. 1).

Podemos comprobar cómo al utilizar este tipo de representación del conocimiento, hemos perdido el significado inicial de la frase, cambiándolo por otro aproximado. Si hubiéramos utilizado otra forma de representación, el significado final de la frase representada habría sido distinto.

Este problema y otros muchos se plantean a la hora de escoger qué tipo de representación del conocimiento utilizar para crear el Sistema Experto.

Condiciones generales de cualquier tipo de representación

Antes de definir los distintos tipos de representación que se manejan hace falta nombrar las condiciones comunes que deben cumplir todos ellos para que sean válidos.

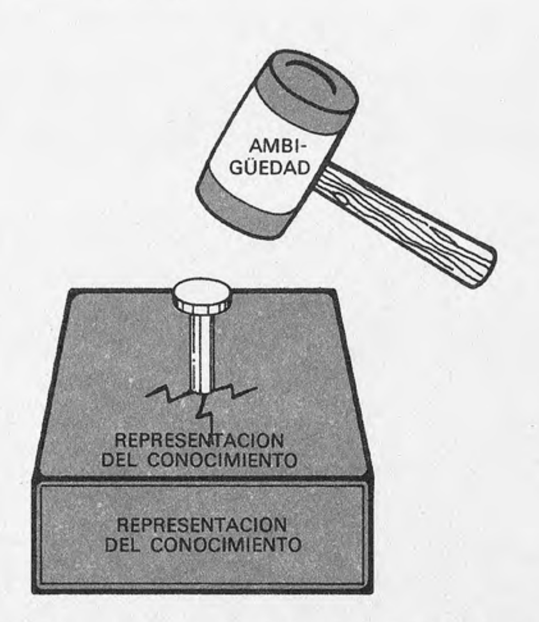


Figura 2.—La ambigüedad es el peor enemigo de la representación del conocimiento.

Es fundamental eliminar la posibilidad de que en algún caso se pueda representar cualquier tipo de ambigüedad, puesto que es éste el principal inconveniente para que una representación se considere aceptable.

Parece claro: si queremos una buena representación del conocimiento no puede ocurrir que estemos representando de una misma manera varios hechos de información distintos. Hay que evitarlo a toda costa. Para ello vamos a analizar los dos tipos básicos de ambigüedad que existen:

- **AMBIGUEDAD REFERENCIAL**

Por ejemplo

"él vino"

puede tener muchos significados distintos en función del contexto en que se encuentre

— primer significado:

"Nacho le ordenó a Luis que viniera y él vino"

En este caso sabemos, por toda la frase, que "él vino" significa que Luis vino.

— segundo significado:

"Nacho le dijo a Luis: vamos. Sin embargo, Luis se marchó y sólo él vino".

En este caso es Nacho el que viene. Vemos cómo la misma frase puede significar cosas distintas en función del contexto en el que se encuentre.

- **AMBIGUEDAD POR EL SENTIDO DE LAS PALABRAS**

Por ejemplo:

"Marcos cogió una pelota"
"Marcos cogió un resfriado"
"Marcos cogió una cogorza"

Aquí podemos comprobar cómo es la palabra la que crea ambigüedad, puesto que el significado de la frase depende de lo que "cogió" Marcos. Esta ambigüedad es muy difícil de eliminar y tal vez sea, precisamente, la que retrase en muchos años el que los ordenadores sean capaces de *entender* todo lo que el hombre

pueda contarle en su propio lenguaje, sin ningún tipo de restricciones.

Tipos de representación del conocimiento

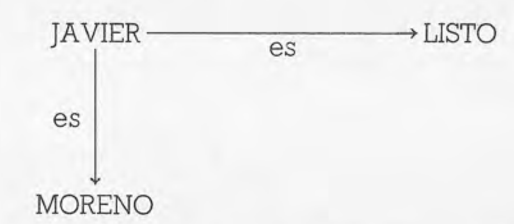
- **REPRESENTACION DE TIPO REDES DE RELACION**

La representación de tipo redes de relación es aquella que utiliza estructuras tipo red (malla) para relacionar los conceptos entre sí.

Por ejemplo para representar las frases:

"Javier es listo" y "Javier es moreno"

se haría:



se puede ver cómo la forma final de representación tiene una estructura tipo red donde los nudos son conceptos y los arcos son la relación entre los conceptos.

- **REPRESENTACION DE TIPO LINEAL**

La representación de tipo lineal no utiliza redes de relación, sino que su representación es estructural.

Por ejemplo, para representar las frases anteriores haría algo del tipo:

para "Javier es listo"
(inst Javier listo)
y para "Javier es moreno"
(inst Javier moreno)

Vemos cómo la representación utilizada tiene una forma lineal en su desarrollo (¡¡Ojo, lineal de línea!!). Aunque hay varios tipos distintos dentro de la forma lineal, todos ellos tienen en común que su estructura no es de tipo árbol de relación, de ahí su definición.

Profundicemos ahora en los dos tipos de representación comenzando por el lineal.

Representación en forma lineal

La representación en forma lineal utiliza cadenas de palabras con una estructura perfectamente definida mediante unas reglas de formación determinadas. Cada cadena recibe el nombre de fórmula.

Las palabras que forman la fórmula son de dos tipos distintos: predicados y argumentos.

Hay un predicado por cada bloque mínimo de información. El predicado enlaza los argumentos para darles sentido en cada fórmula. Además, da el sentido que debe tener la fórmula.

Por ejemplo, la frase:

"María es alta"

se representaría,

(inst. María alta)

donde "inst" (que proviene del inglés instance) es un predicado que enlaza los argumentos María y alta. Su función es definir que María es alta, es decir, relacionar "María" con "alta".

La ventaja de esta estructura es que una fórmula ya compuesta, como por ejemplo (inst. María alta), puede formar parte de otra fórmula más complicada como argumento. Así se consigue una gran flexibilidad y se permite la construcción de estructuras complicadas a partir de otras más simples.

En relación con toda esta definición, vamos a analizar un sistema muy utilizado para trabajar con el ordenador que particulariza las ideas generales que hemos expuesto hasta ahora: es el "cálculo de predicados". Posee sus propias normas particulares, aunque la base nos resultará conocida.

Cálculo de predicados

Es un método muy utilizado para la representación de proposiciones y para el posterior manejo que de ellas se puede hacer para realizar inferencias.

Básicamente consiste en crear células básicas de información (fórmulas), constituida cada una de ellas por predicado y argumentos.

El ejemplo anterior sirve a su vez como definitorio del cálculo de predicados:

(inst María alta)

inst → Predicado
María → Argumento
Alta → Argumento

Para evitar la ambigüedad al definir los elementos se asignan coeficientes numéricos para conseguir que cada argumento tenga un único significado.

Por ejemplo:

(inst Silla-1 Silla)	Silla-1 es una silla
(inst Silla-2 Silla)	Silla-2 es una silla

Silla-1 y Silla-2 son definidas como sillas, pero puede ocurrir que una sea clásica y la otra tipo taburete de bar; precisamente para diferenciarlas se les asignan coeficientes numéricos distintos.

- **ARGUMENTOS**

Hay tres tipos distintos de argumentos que se pueden utilizar en las fórmulas:

— Símbolos constantes
— Variables
— Funciones de aplicación

Los símbolos constantes pueden ser palabaras que haya que relacionar.

Por ejemplo, 'Juan', 'Hombre' o cualquier otro.

Las variables tienen el mismo papel que en matemáticas, es decir, pueden tomar cualquier valor dentro del intervalo de definición que posean.

Por ejemplo, si tenemos:

(inst X Hombre)

lo que se consigue es que la variable "X" pueda tener el valor "hombre".

Las funciones de aplicación son, como ya hemos comentado anteriormente, las fórmulas complejas que sirven de argumento en otras.

Por ejemplo, una fórmula compleja que puede utilizarse como argumento podría ser:

(inst Pedro cojo)

- **PREDICADOS**

Como el cálculo de predicados no es un lenguaje en el que las palabras "clave" estén definidas previamente, y no hay que utilizarlas forzosamente, se permite una gran riqueza a la hora de la representación.

Esto significa que podemos utilizar como predicados los que nos interesen para representar lo que queremos.

Por ejemplo, podemos tener:

(hijo-de Marta María)

queremos representar que Marta es hija de María y utilizamos el predicado "hijo-de".

En realidad el cálculo de predicados sirve de soporte para programar en los lenguajes que maneja el ordenador (puesto que todavía estamos en el siglo XX). Estos lenguajes, de los que el más común en I.A. es el LISP, están estructurados para trabajar con información alfanumérica, y el cálculo de predicados se amolda mucho a ellos.

Como subrango del papel que realizan los predicados, tenemos los elementos conectores y relacionadores de fórmulas.

Existen dos básicamente:

CONECTORES (if, and, or, not)

tienen la facultad de permitir conocer la certeza o falsedad de la fórmula que forman conociendo la del argumento sobre el que actúan.

Por ejemplo

(not (inst coche rojo))

El significado de esa representación es "el coche no es rojo".

Si analizamos el valor podemos ver si el coche que queremos representar *realmente* es rojo, en cuyo caso
(inst coche rojo) es cierto.
y sabiendo esto sabemos también que
(not (inst coche rojo)) es falso, puesto que el coche es realmente rojo.

CUANTIFICADORES

Hay dos

— Existencial, simbolizado como ∃ (exists)
tiene en el cálculo de predicados un sentido similar al que poseía en matemáticas (existe...)

— Universal, simbolizado como ∀ (forall)

se representa de la forma siguiente:

(forall (X) A)

y significa que la variable X está cuantificada de forma universal en la fórmula A. Por ejemplo: (forall (hombres) han nacido) indica que todos los hombres han nacido.

Reglas del conocimiento

Cuando estudiamos la jerarquía que existía dentro de los niveles de información comentamos que el conocimiento se conse-

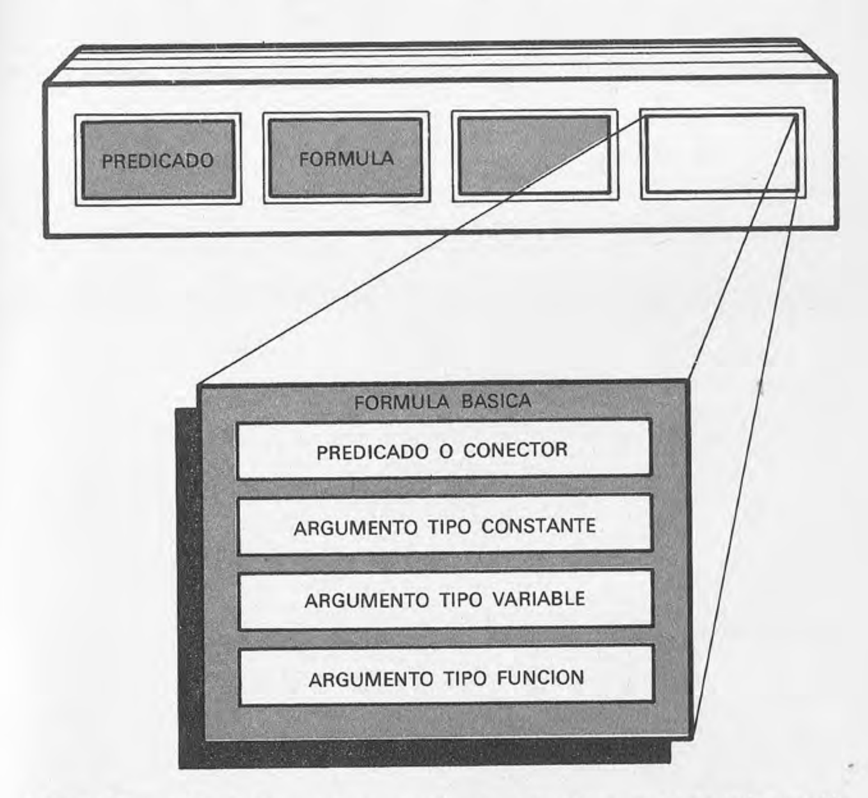


Figura 3.—La figura esquematiza la estructura de toda información representada mediante cálculo de predicados.

guía precisamente mediante las "reglas del conocimiento". Pues bien, estas reglas deben de poder representarse en el cálculo de predicados puesto que, en caso contrario, no serviría de nada esta teoría. Y, en efecto, se pueden representar.

Como existen muchas formas de hacerlo nosotros nos hemos centrado aquí en las reglas de producción. Su estructura es la típica de cualquier regla:

IF (antecedente) THEN (consecuente)

donde IF y THEN son los conectores ingleses que corresponderían a:

SI (antecedente) ENTONCES (consecuente)

Por ejemplo

" si llueve entonces me mojo"

La forma de representarla mediante el cálculo de predicados tendría el formato siguiente.

(if P Q)

el significado de esta expresión sería: 'si es cierta la proposición P entonces es cierta la proposición Q'.

Reglas de control

Ya hemos visto las reglas de producción. Ahora lo único que falta es saber cómo se va a realizar la inferencia y qué se logrará inducir a partir de unas reglas del conocimiento iniciales.

Esto se consigue definiendo para cada sistema las reglas de control que lo gobiernan, también conocidas como reglas de inferencia. Trabajan con los hechos básicos y las reglas de producción para poder deducir, inducir, etc.

Vamos a analizar algunas de las que existen en las matemáticas lógicas.

- **MODUS PONENS (DEDUCCION)**

La regla de inferencia dice así: dados como información cierta

(if P Q)
y
P

se consigue tener la certeza de Q.

Lo podemos representar como

(if P Q) }
P } ——→ Q

Por ejemplo: si tenemos como información inicial

(si llueve entonces me mojo)
y
(hoy llueve)

podemos tener la seguridad de que "hoy me mojo"

Existe una regla que se puede concluir de la "modus ponens": Es la 'modus tollens'.

Dice así:

dados: (if P Q)
y
$\overline{Q}$

se obtiene $\overline{P}$. La barra situada sobre la $\overline{P}$ y la $\overline{Q}$ significan NOT (NOT P y NOT Q)

- **ABDUCCION**

Es una regla de inferencia muy buena para generar el proceso de explicación que se le pide al sistema experto.

Funciona así:

Dados (if A B)
y
B

Se saca A.

Tiene asociado un problema fundamental, y es que puede llevar a conclusiones falsas. Por ejemplo de:

"Nacho tiene dolores"
y
"Si enfermas de cáncer entonces tienes dolores"

se concluye que "Nacho tiene cáncer", lo cual es como mínimo muy aventurado.

Sin embargo, tiene un gran valor por su avanzado nivel de riesgo.

• INDUCCION

Esta regla de inferencia es muy buena en los procesos de aprendizaje del sistema experto. Su significado es el mismo que posee en las matemáticas clásicas, es decir, se basa en los procesos de generalización matemática a partir de unos ciertos elementos individuales. Por ejemplo, si tenemos la certeza de:

"Juan respira"
y
"María respira"

Se puede intentar generalizar que

"Todos los hombres respiran"

Es también una regla arriesgada, puesto que fácilmente se incurre en errores importantes al generalizar características que pertenecen sólo a elementos aislados.

Se podría esquematizar de la siguiente forma:
si definimos

(p A)

(cuyo significado sería que 'A' cumple la propiedad 'p'), al tener la certeza de (p A) y (p B) podemos inferir:

(forall (X) (p X))

que significa que para todo X se cumple la propiedad 'p'. Además de estas reglas de inferencia existen otras que también se utilizan en los sistemas expertos. Con todas ellas se puede conseguir un buen motor inferencial; en función de las que se escojan para crearlo el motor tendrá unas u otras características.

Representación tipo redes de relación

Es la segunda gran tendencia para la representación del conocimiento.

Se utilizan las estructuras en red como método de relacionar conceptos e información.

La red se compone de nudos y arcos. Los nudos representan conceptos y los arcos son los lazos de unión entre ellos:

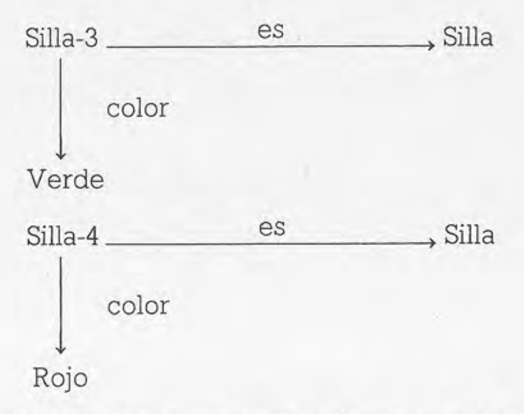
A —relación→ B

Por ejemplo:

Javier —aprueba→ estadística

En estos tipos de representación también hay que evitar la ambigüedad. Para ello se utilizan coeficientes numéricos diferenciadores.

Así:



Podemos observar dos cosas en los ejemplos anteriores: que es necesario que existan coeficientes anti-ambigüedad (en este caso diferencian dos sillas distintas) y que cada nudo puede ser punto de encuentro de varias áreas; con esto se consigue crear una estructura tan compleja como se desee y permite representar conceptos complicados.

Ahora habría que analizar todo el conjunto de herramientas que, actuando sobre este tipo de representación del conocimiento, facilitan la creación de alguna estructura que represente las famosas *reglas de producción* de la representación lineal. Asimismo también es necesario en este caso controlar los procesos de inferencia que tengan lugar en el sistema.

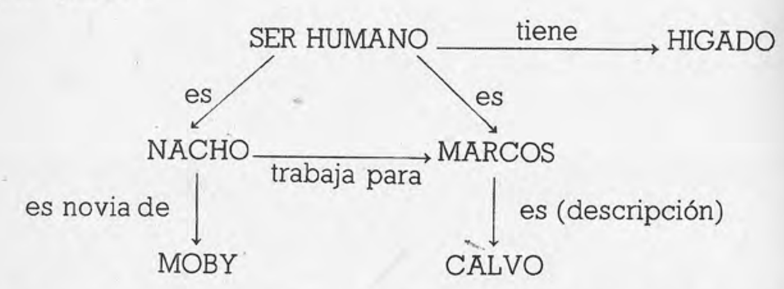
Para realizar todo este control existen unas técnicas de gestión de grafos y unas propiedades (también de grafos) que permiten un manejo muy adecuado de la base de conocimientos. No las expondremos aquí en profundidad, sino que nos limitaremos a definirlas.

Algunas de estas herramientas para la gestión de la red son:

• **HERENCIA POR JERARQUIAS**

Las propiedades que posee un concepto las adquieren por 'herencia' todos los 'hijos' de dicho concepto.

Por ejemplo:

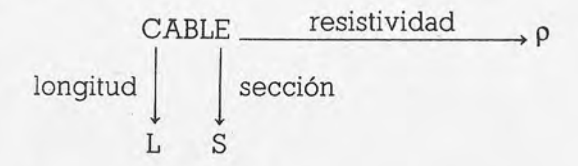


Todo ser humano tiene hígado. Como Marcos es un ser humano, también tiene hígado y no hace falta colgar un arco directamente del nudo MARCOS al nudo HIGADO. Lo mismo ocurre con NACHO.

- **DEMONS**

Se utiliza como procedimiento para hallar valores o resultados partiendo de información ya existente. Supongamos que queremos hallar la resistencia eléctrica de un cable de alta tensión. Podremos hacerlo si conocemos el tipo de material que forma el cable, su resistividad, la longitud del cable y su sección. Si tenemos todo eso representado en una red podremos calcular otra propiedad del cable como es la resistencia eléctrica.

Así se tendría:

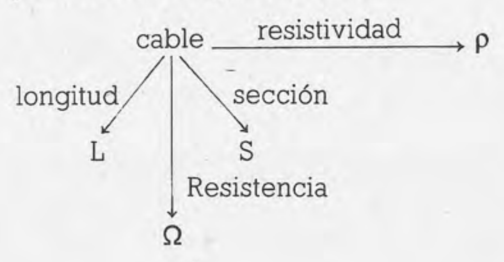


El proceso a seguir debería ser:

1) Ver si existen sobre el nudo cable las relaciones resistividad, sección y longitud.
2) En caso de que existan, crear una nueva relación donde el valor que le corresponde se consiga mediante la multiplicación y división de los valores que posean las anteriores, de la forma siguiente:

$$\text{Resistencia} = \text{Resistividad} \times \frac{\text{longitud}}{\text{sección}}$$

El grafo quedaría de la siguiente forma:



- **DEFAULT**

Se utiliza para permitir, en cierta medida, el trabajo de la red con un concepto nuevo: la probabilidad.

Hasta ahora hemos hablado del almacenamiento y gestión del conocimiento sin analizar la posibilidad de que éste no fuera del todo cierto o bien fuera inexacto.

Aunque los problemas de certeza y precisión del conocimiento son muy importantes, aquí simplemente haremos referencia a ellos y no los trataremos con profundidad. En el caso de que el lector pretenda profundizar algo más en este área encontrará una bibliografía al final del libro donde encontrará información al respecto.

El Default (palabra que proviene del inglés y significa "por defecto") se utiliza para facilitar el manejo de información que no posee un valor de veracidad absoluto.

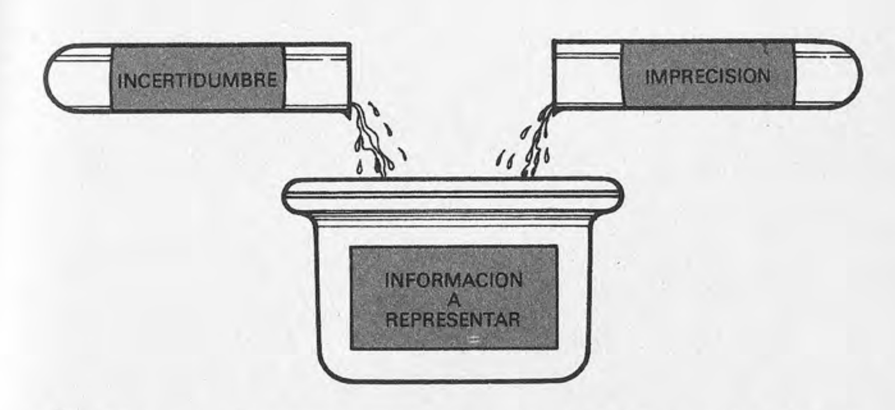


Figura 4.—Toda información posee dosis de imprecisión e incertidumbre que dificultan, y pueden llegar a hacer imposible, su utilización eficaz.

Por ejemplo:

Juan $\xrightarrow{\text{color-pelo (default)}}$ rubio

quiere decir que el pelo de Juan probablemente sea rubio.

Existen otros problemas aparte de la probabilidad de verdad en una información; ya hemos citado la certeza y la precisión. Debido a estos problemas la representación del conocimiento no ha podido evolucionar tan rápido como en un principio se preveía.

CAPITULO IV

ESTUDIO DEL MOTOR DE INFERENCIA

Hemos visto ya la problemática básica de la representación del conocimiento, por qué se necesita un tipo determinado de representación y los múltiples problemas que aparecen al estudiar el tema. Respecto al motor de inferencia lo único que hemos comentado es que su principal misión era seleccionar las reglas de producción que debían utilizarse para resolver cada problema propuesto por el usuario y cómo el motor de inferencia realizaba sus inducciones mediante unas reglas de control determinadas.

Ahora vamos a analizar un poco más profundamente cómo funciona un motor de inferencia "por dentro".

Lo primero será estudiar el funcionamiento del motor cuando debe actuar, en sus dos fases: evaluación y ejecución.

Una vez que se profundice en el estudio del motor se explicarán los pasos necesarios para que su funcionamiento sea correcto y ajustado al que se desea, que concretamente, son: la inferencia, el concepto de búsqueda, el control de la operación y el control del espacio de búsqueda.

Fases de operación del motor

Para resolver un problema mediante un sistema experto hace falta recorrer todo un camino de inferencias hasta llegar a una solución final; a lo largo del mismo hace falta que el motor escoja muchas reglas de conocimiento que están almacenadas en su base de conocimientos.

Cada vez que hay que elegir una de esas reglas debe de ponerse en funcionamiento toda una compleja estructura que permita escoger la mejor, aplicarla, etc. Esto lo debe hacer el motor de inferencia un gran número de veces.

Para permitir optimizar este proceso el motor está diseñado para que funcione por ciclos. Primero realiza una tarea y luego otra.

Las principales fases del motor de inferencia son:

— Evaluación
— Ejecución

• EVALUACION

Como su nombre indica, en esta fase el motor se ocupa de escoger las reglas que debe aplicar para cada problema de entre todas las reglas que existen en su base de conocimientos.

Esto, que parece tan fácil a primera vista, resulta muy complicado, con problemas de difícil solución, como, por ejemplo, los conflictos entre reglas, que analizaremos posteriormente.

La primera etapa de esta fase, como se puede ver en la figura 1, consiste en la restricción. Se trata de recoger todas las reglas que, por su estructura, se puedan aplicar a un determinado problema planteado. De entre todas las reglas (R) las elegidas formarán un subconjunto R_1.

Posteriormente hay que realizar un filtrado de las reglas seleccionadas eligiendo las que encajan perfectamente con el problema planteado. Se consigue el subconjunto R_2.

Una vez que se ha conseguido el conjunto de reglas R_2 hay que eliminar de ese subconjunto las reglas que provoquen conflictos. Este paso es importante. Supongamos que estamos analizando nuestro coche con el "maravilloso" "cuida-coches electrónico" y supongamos también que en R_2 tenemos dos reglas que dicen:

R_1 — "Si el coche hace ruido al acelerar entonces hay que parar inmediatamente."
R_2 — "Si el coche hace ruido al acelerar entonces hay que acelerar más todavía."

Podemos imaginar que el ordenador "se armaría un lío" si tuviera estas dos reglas para aplicar al problema propuesto. Por eso es necesario resolver todos los conflictos que aparezcan en R_2.

Las dos últimas etapas son las que están menos desarrolladas actualmente y es aquí en donde flaquean más los sistemas expertos.

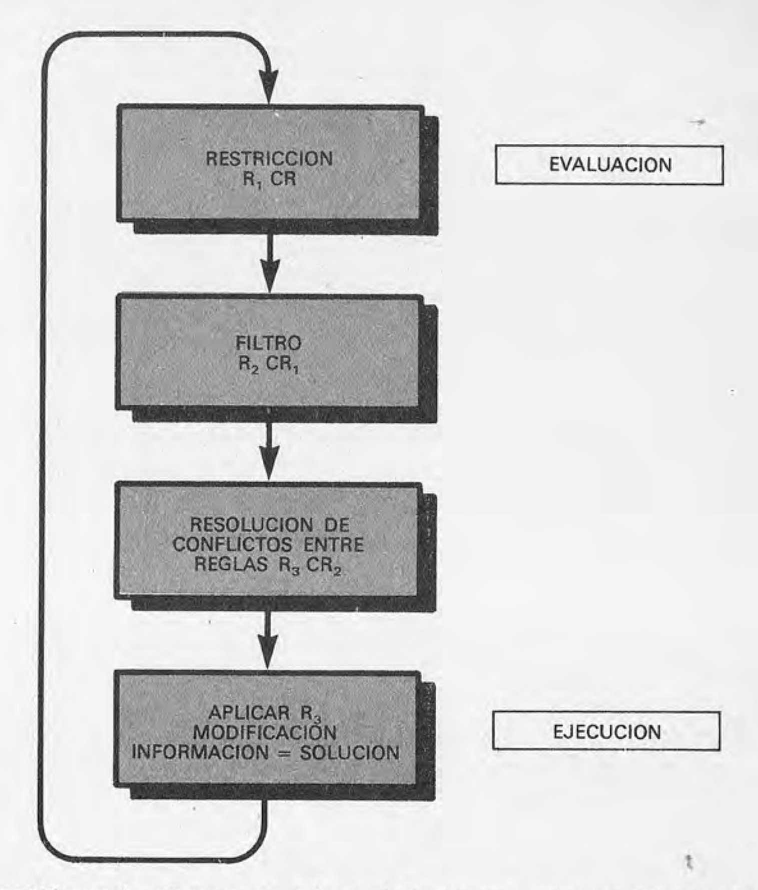


Figura 1.—El motor de inferencia funciona mediante ciclos, con unas fases perfectamente definidas que van estrechando el conjunto de reglas adecuadas para el problema.

Los bucles de inferencia son lazos cerrados de inducción: supongamos que tenemos 4 reglas A, B, C, D y que al activar "A", ésta nos manda a "B", "B" a "C", "C a "D" y, por fin "D" a "A". Tendríamos lo señalado en la figura 2.

Esto obligaría al ordenador a inferir siempre lo mismo, dentro de un bucle sin fin. Este problema también debe ser detectado y eliminado para que el sistema funcione correctamente. Para evitar los "bucles de inferencia" hace falta una herramienta muy potente.

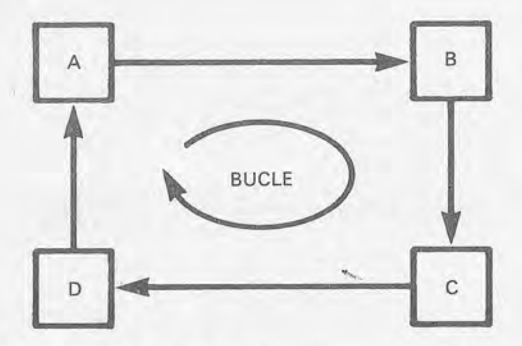


Figura 2.—Los bucles de inferencia son lazos cerrados de inferencia en los sistemas expertos. Hay que evitarlos a toda costa.

• **EJECUCION**

Esta fase se ocupa de que las reglas escogidas en la fase anterior (suele ser una regla sola) se ejecuten, produciendo un cambio en la información que se presentará como solución del problema propuesto.

Una vez realizada la ejecución el sistema comprobará si existe una condición de final y, en caso contrario, pasará a una nueva fase de evaluación.

Funcionamiento del motor de inferencia

Ya sabemos cuáles son las fases que lleva a cabo el motor. Ahora vamos a analizar cómo se hace realmente todo lo que hemos explicado anteriormente.

Existen dos tareas que debe realizar el motor.

- Inferencia
  Se debe de aplicar la inferencia al ejecutar una de las reglas. Estas se ejecutan en función del tipo de inferencia que posea el sistema experto.
- Control
  El control se ocupa de toda la gestión del sistema que hemos analizado anteriormente; tanto de la forma en que se escogen las reglas como de la resolución de conflictos o de la eliminación de bucles de inferencia.

Es importante el papel que juega el usuario en esta fase en la mayoría de los sistemas, pues muchas veces el ordenador no conoce algunos datos importantes para gestionar el sistema y entonces se los pedirá al usuario. De esta forma se puede suplir la falta de información.

Dentro de la fase del control hay que distinguir dos tipos:

— *Control del espacio de búsqueda*
Se ocupa de gestionar "cómo se escogen las reglas" en función del espacio de búsqueda concreto (conjunto de reglas que se pueden aplicar dependiendo de cada problema).

— *Control de la operación*
Se ocupa de resolver conflictos y realizar todas las tareas para gestionar la fase de evaluación.

• **INFERENCIA**

Para realizar las inferencias hay que tener almacenadas unas determinadas reglas de control que enseñen al sistema cómo llevarlas a cabo.

Las reglas de control (llamadas también reglas de inferencia) las hemos estudiado ya y no las repetiremos; corresponden al Modus Ponens, Abducción, etc.

Aquí se puede comprobar dónde se aplican las reglas que se analizaron anteriormente.

• **CONTROL**

Ya hemos definido varias veces las tareas de control de un sistema experto para que funcione bien. Este debe de enfrentarse con dos problemas básicos, consistentes en disponer de métodos para:

- decidir dónde empezar a analizar las reglas que se almacenan en la base de conocimientos;
- resolver los conflictos entre reglas y los bucles de inducción.

Además de resolver estos problemas, es necesario que el sistema trabaje óptimamente gestionando el espacio de búsqueda de una forma adecuada.

ESPACIO Y ARBOL DE BUSQUEDA

Cuando en etapas anteriores definíamos el motor de inferencia como el elemento del sistema que debía escoger las reglas que se tenían que aplicar en cada momento, no se comentó nada

de cómo se conseguía esta elección; el proceso es realmente complicado.

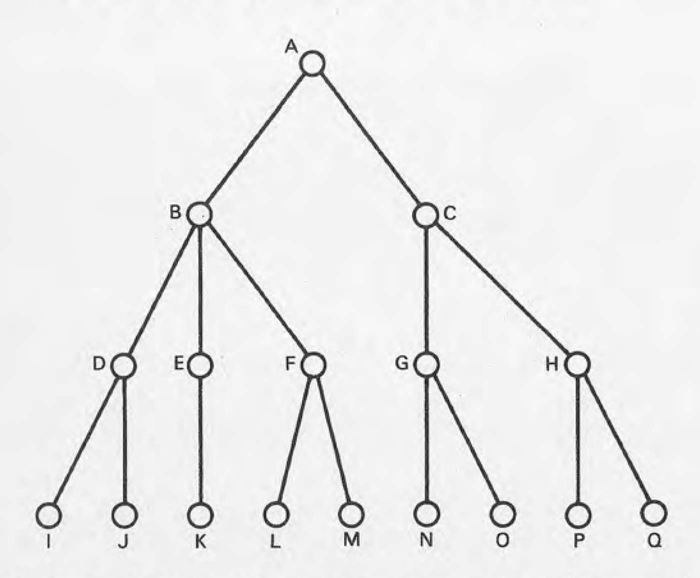
Como sabemos, el ordenador no puede inventar nada. Todo lo que haga debe haber sido programado previamente.

Por tanto, los métodos de elección de reglas no son "inteligentes" tal y como nosotros lo entendemos.

Para evitar el problema que esto representa se ha creado toda una teoría de modelización que intenta esquematizar al máximo todo el problema.

Así se crea inicialmente el concepto de espacio de búsqueda. Cuando debido a la aparición de un problema se activa la primera regla para resolverlo, ésta podrá permitir la aplicación de un determinado número extra de reglas y cada una de ellas permitirá la aplicación de otras.

Así se crea un árbol (grafo 1), que representa el conjunto de posibles aplicaciones de reglas. El problema quedará resuelto cuando a lo largo de todo el árbol (árbol de búsqueda) se consiga encontrar un camino que nos lleve a la solución.



Grafo 1.—Arbol de búsqueda genérico, con las reglas y subreglas situadas en los distintos modos.

En función de la regla elegida inicialmente se tiene un árbol diferente. El árbol completo para un problema determinado normalmente suele resultar extenso.

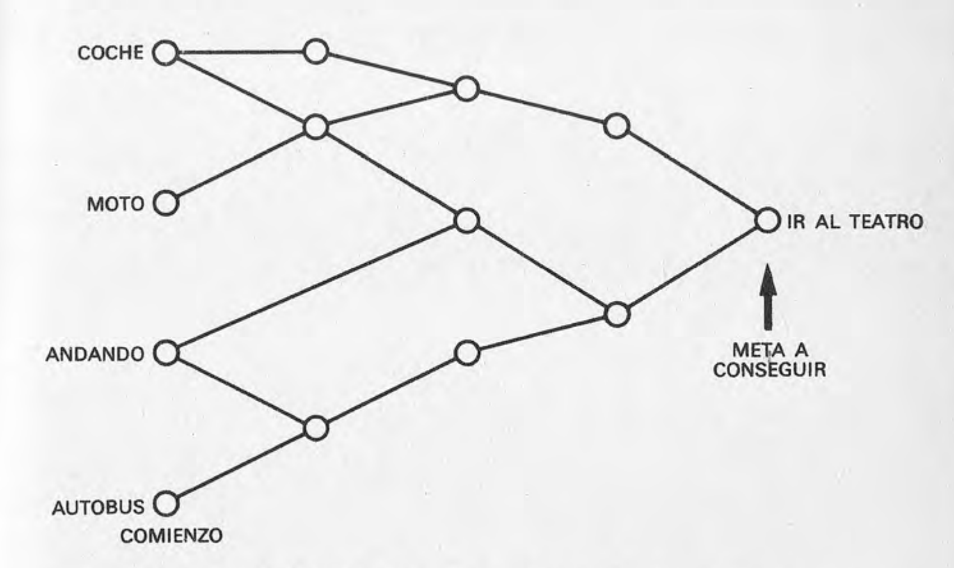
Gestión de los árboles de búsqueda

Cuando ya se posee el árbol de búsqueda para la resolución del problema aparecen muchas formas de gestionarlo. Para elegir entre ellas se suelen realizar estudios sobre la forma que posee dicho árbol, distinguiéndose dos tipos básicos en función de los problemas que hay que resolver.

Supongamos que el problema a resolver es que queremos ir al teatro. Para ir al teatro hay muchas formas posibles: en coche, andando, en autobús, etc.

Lo que debemos hacer es escoger entre las múltiples formas la que más nos convenga en función de la distancia, el clima, el precio, etc.

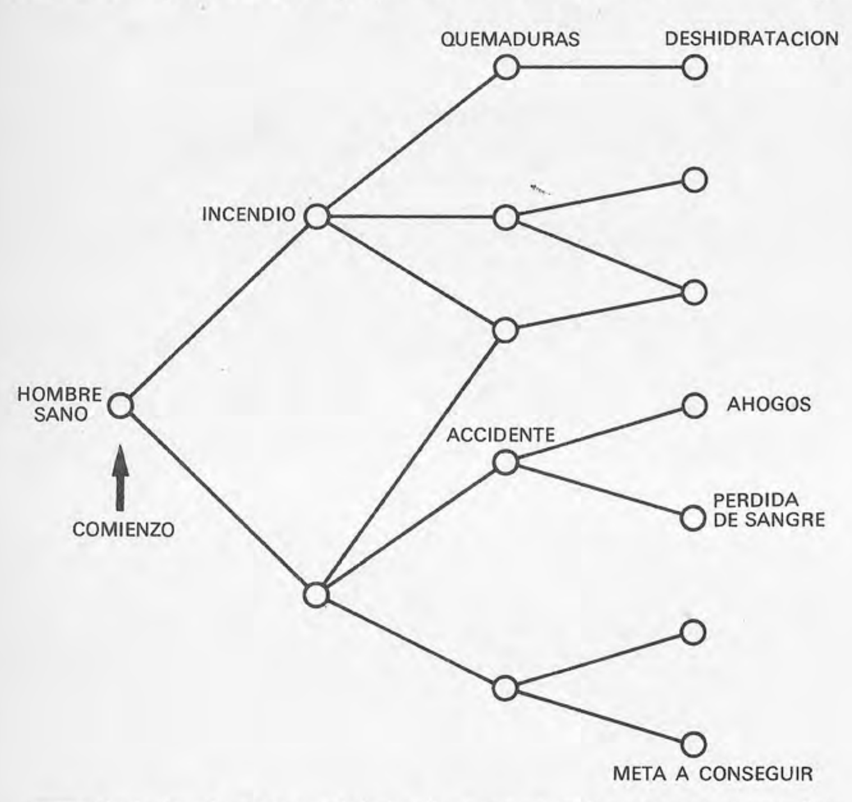
El árbol de búsqueda que tendremos será del tipo mostrado en el grafo 2.



Grafo 2.—Posible árbol de búsqueda para "ir al teatro", perteneciente al grupo de los que van reduciendo posibilidades al avanzar en ellos.

Existen otros problemas, sin embargo, en los cuales el árbol de búsqueda es completamente distinto. Por ejemplo, cuando queremos predecir algo pero no sabemos concretar, puesto que depende de los datos iniciales que se posean.

Se trata de inferir posibles consecuencias de los hechos. El árbol de búsqueda sería algo parecido al del grafo 3.



Grafo 3.—En algunos árboles lo que se produce es una mayor "ramificación" según profundizamos.

Estos dos árboles dan pie a distintas maneras de gestionarlos. Cuando como en el primer árbol, se parte del objetivo a cumplir y se busca la manera de realizarlo, al método se le llama encadenamiento desde atrás. Parte del final para llegar al principio.

Sin embargo, si, como en el segundo método, se parte del comienzo y se deja evolucionar el sistema hasta que llega al objetivo, se llama encadenamiento desde delante.

En función del tipo de problema que se vaya a resolver habrá que utilizar una de estas dos teorías para gestionar la elección de reglas correspondiente.

Esto significa que estos dos métodos son formas de plantear el problema y no políticas de elección de reglas.

Un camino en el árbol: elección de reglas

Una vez escogida la manera de gestionar el problema se planteará la política de elección de reglas que se desea. La elección de reglas representa encontrar un camino sobre el árbol de búsqueda. Y, claro está, hay muchas formas de conseguir un camino (escoger reglas), cada una de ellas diferentes.

Se diferencian generalmente dos formas de realizar la elección.

- **Escoger las reglas de una manera mecánica**

Se crea una función predefinida que escoge las reglas siempre desde un mismo punto de vista.

- **Escoger las reglas de forma más inteligente**

Generalmente se utilizan métodos de evaluación del estado en que se encuentra el sistema de forma que en función de la elección de una regla determinada se alcance más rápidamente la solución. Muchas veces se aplica la experiencia acumulada de otros procesos. Son los llamados métodos heurísticos.

ELECCION MECANICA

Para conseguir un camino de forma mecánica en el árbol de búsqueda que representa al problema planteado existen dos formas básicas.

BUSQUEDA EN PROFUNDIDAD

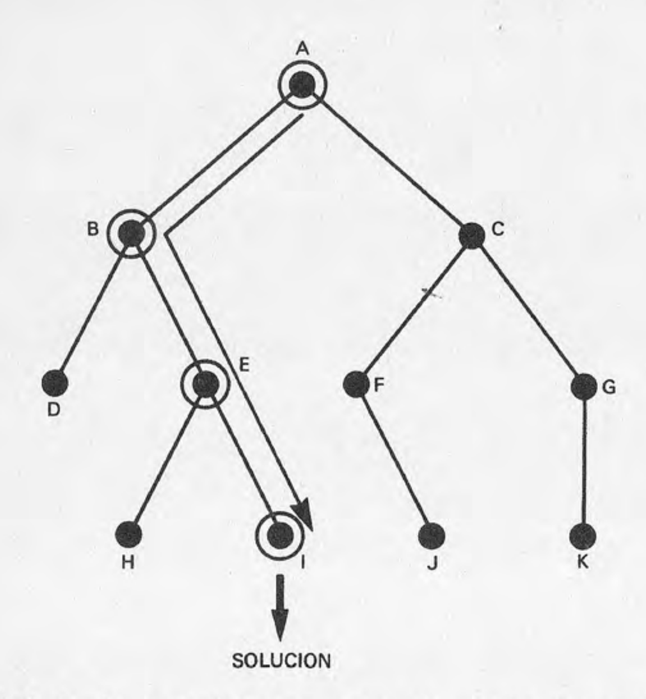
Como su nombre indica, se trata de partir del nudo inicial y buscar la solución al problema profundizando lo máximo posible en el árbol de búsqueda. Cuando se llega a un punto donde no se puede continuar entonces se retrocede para buscar un nuevo camino en profundidad.

Veamos un ejemplo aclaratorio. Supongamos que, viendo el árbol del grafo 4 hemos resuelto el camino para conseguir la solución.

De la figura se deduce que si aplicamos las reglas A, B, E e I (en ese orden) llegamos a solucionar el problema.

Una búsqueda en profundidad buscaría la solución rastreando el árbol de la siguiente forma. (ver grafo 5): empezaría por A, continuaría escogiendo un nudo descendiente del A, podría ser el B o el C. Toma el B de una forma arbitraria.

Una vez escogido el B se analiza si se ha llegado a la solu-



Grafo 4.—Camino válido para alcanzar la solución (y óptimo).

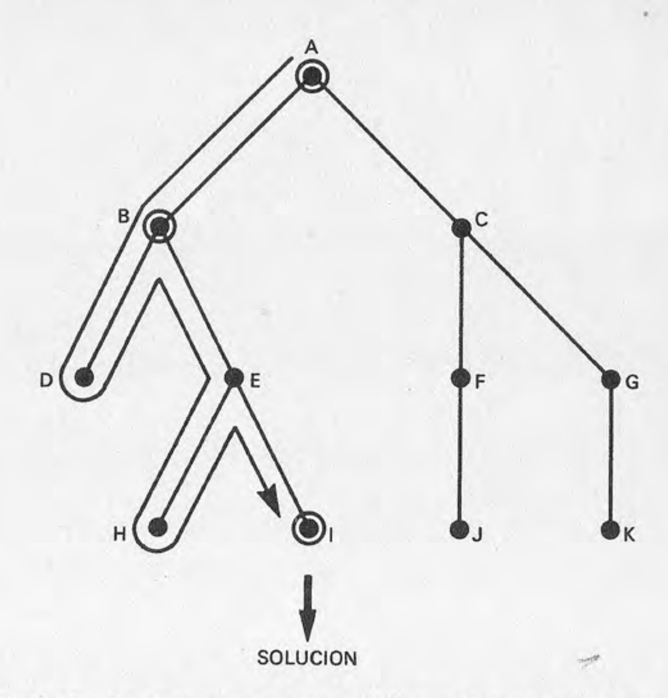
ción, que se encuentra en el nudo I. Como no se ha llegado se profundiza hasta D. Al llegar a D, que tampoco es solución, no se puede continuar; por lo tanto se vuelve a B para pasar posteriormente a E. Después a H y una nueva vuelta a E para por fin llegar a la solución.

Este es el método de búsqueda en profundidad.

BÚSQUEDA EN EXTENSION

Va recorriendo todos los nudos de cada nivel. Ante el mismo problema, planteado en el mismo árbol de inferencia, una búsqueda en extensión actuaría de la siguiente forma (representada en el grafo 6)

Se partiría inicialmente del nudo A; como en el caso anterior, se pasaría al nudo B. Una vez analizado que no se ha llegado a la solución, en vez de continuar en profundidad se continua "en extensión", analizando el nudo C. De C, como no existe otro nudo "al mismo nivel" se pasa al nudo G. Así se van analizando todos los



Grafo 5.—Proceso seguido en la búsqueda mecánica en profundidad para alcanzar la solución.

nudos del mismo nivel hasta D. Se baja hasta H y de H se pasa a I, la solución final.

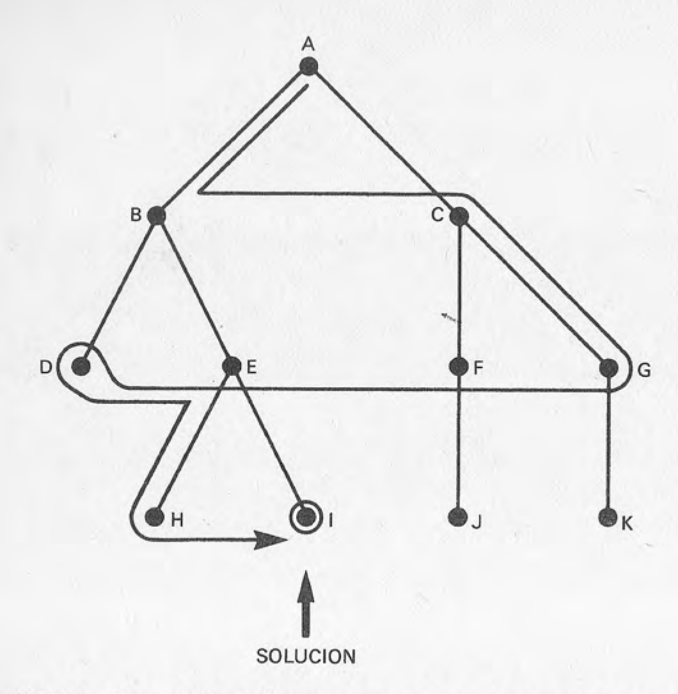
El escoger uno de estos dos métodos depende, como en el caso de las estrategias de búsqueda, del tipo de problema que se pretenda solucionar.

Lo que ocurre normalmente es que los sistemas expertos se diseñan sólo con un tipo de método, por lo cual adquieren peculiaridades que les hacen específicos para un tipo de problemas determinado.

ELECCION HEURISTICA

Al fijar una forma mecánica de escoger las reglas (por ejemplo, en profundidad) se puede comprender que el sistema no posee ningún tipo de conocimiento sobre si durante su operación se está acercando a la solución o bien se está alejando.

Simplemente toma una regla y comprueba si ha llegado al fin



Grafo 6.—Sistema de búsqueda mecánica por extensión.

que se pretende; en caso negativo escoge una nueva regla en función de la norma que tiene fijada.

Lo que pretenden los métodos heurísticos de elección de reglas es precisamente dar información al sistema acerca de cómo se encuentra en cada instante y, lo que es más difícil, sobre cómo quedaría aplicando una determinada regla. Una vez que se posee esa información, el sistema puede escoger la regla que optimice su operación.

Por ejemplo, si se consiguiera saber exactamente cómo se comporta el sistema ante el problema que propusimos anteriormente, la elección de reglas sería precisamente la que diera el camino directo desde el estado inicial hasta la solución, es decir, se aplicarían las reglas A, B, E, I por ese orden. Podemos comprobar cómo este método utiliza el mínimo número de reglas. Es el más rápido y preciso.

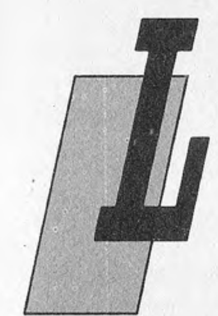
Como realmente no existe un método capaz de conocer perfectamente el problema, se utilizan aproximaciones que lo

que permiten es reducir al mínimo los errores en la elección de reglas.

Estas aproximaciones son funciones o algoritmos que coloca el ingeniero del conocimiento y que él debe escoger. En este camino se encuentran los últimos adelantos en inteligencia artificial que se realizan en Estados Unidos, Japón y Europa.

CAPITULO V

PROCESADORES DE LENGUAJE NATURAL



Las necesidades de información y manejo informático a los que están sometidos la mayoría de los profesionales en los más diversos campos está provocando un cambio sustancial en la relación hombre-ordenador. No es el especialista en informática el que tiene que contactar con la máquina, sino estudiosos de otros temas, normalmente usuarios inexpertos en su manejo.

La inteligencia artificial ha permitido diseñar una serie de herramientas, los *procesadores del lenguaje natural,* que faciliten la interrelación hombre-ordenador y permitan una comunicación mucho más fluida y menos estructurada que los sistemas tradicionales de menús.

Los procesadores de lenguaje natural están compuestos funcionalmente de un *analizador,* un *diccionario* y un *direccionador.* El analizador estudia, bien sintáctica o semánticamente, la estructura de las oraciones con ayuda de reglas incorporadas al sistema. El diccionario expresa relaciones entre conceptos y almacena palabras, siendo el direccionador el encargado de pasar del lenguaje natural analizado a un lenguaje formalizado, propio del sistema donde se está trabajando.

Necesidad del lenguaje natural en la comunicación con el ordenador

Desde el momento en que el ordenador ha empezado a introducirse en la industria, los servicios públicos e incluso en nuestros hogares, surge la necesidad de facilitar al usuario la comuni-

cación con el mismo. Desde el inicio de la era del ordenador una de las grandes tendencias ha sido la de facilitar el trabajo con las máquinas, siendo éstas las que lleven el mayor peso posible en dicha interrelación. Es hora de que se desarrollen paquetes de software que permitan al usuario consultar por sí mismo, sin muchos conocimientos previos y en su lenguaje natural, y que sea el ordenador el que realice la tarea de traducirlo al lenguaje formal de la propia máquina.

Si bien el campo del lenguaje natural tiene una accidentada y triste historia, la inteligencia artificial y las herramientas que utiliza pueden ser la tecnología que hacía falta para un funcionamiento satisfactorio de los interfaces. Es necesario resaltar desde el principio que los procesadores de lenguaje natural (PLN) son todavía muy limitados en sus aplicaciones para no dejarse inundar por un excesivo optimismo. Los problemas inherentes a la naturaleza del lenguaje, tales como su *escasa lógica* o sus *múltiples estructuras*, no están ni mucho menos resueltos y existen muchos detractores de su utilización.

La aplicación más inmediata de los procesadores de lenguaje natural es la recuperación de información. A la vez que se desarrollan sofisticados métodos de manejo de información, tales como los sistemas inteligentes de recogida, catalogación y recuperación automática, se vio la necesidad de mejorar también el interface con dichas bases de datos. Si se consigue enseñar al ordenador lo suficiente del lenguaje natural como para que sea capaz de entender las preguntas planteadas, se eliminan dos grandes inconvenientes:

- El intermediario humano. Es difícil que el intermediario comprenda realmente lo que el usuario quiere preguntar debido, fundamentalmente, al distinto conocimiento del tema que poseen ambas personas;
- Tener que enseñar al usuario, profesional en otros campos, los lenguajes propios del ordenador.

Funciones que realiza un procesador de lenguaje natural

Los procesadores de lenguaje natural son capaces de realizar una variada serie de funciones:

- Permiten al usuario tener un menor conocimiento del sistema, debido a que evita la utilización de estructuras fijas.
- Corrigen errores de tipo léxico. Al reconocer una palabra que contiene una falta, el programa intentará asociarla con alguna del diccionario. Si alguna de éstas encaja lo suficiente confirma la corrección del fallo y la frase es analizada. Ladeer, sistema experto sobre información naval, nos proporciona un ejemplo:

  * ¿Cuánto dista Kitty HwK de Gibraltar?
  Corrección   Hawk
  Analizado.

- Permitir al usuario construir sus frases de manera incremental, refiriéndose a frases anteriores. Esta capacidad se denomina anáfora y existen dos tipos:

  * Sustitución y elipsis. El PLN puede sustituir el significado de alguna palabra previa (sustitución) o bien entender una pregunta en la que no aparece ninguna referencia previa (elipsis).
  * ¿Qué coches corren a más de 100 Km/h.?
  * ¿Cuáles de ellos entre 100 y 150 Km/h.? (sustitución)
  * ¿Cuáles entre 150 y 200 Km/h.? (elipsis)
  * Referencias a pronombres. Un pronombre, a diferencia de una sustitución, se suele aplicar a valores anteriores más que a significados.
  * ¿Por qué cortaste la madera?
  Para hacer una cama
  * ¿Cómo lo hiciste?

- Responder a preguntas de forma inteligente.
  A la pregunta:
  ¿Está Begoña matriculada en octavo curso?
  (Y el octavo curso no existe), la respuesta puede ser:
  * No
  * No, no existe el octavo curso (mucho más clara y precisa).

- Permitir la ampliación de las reglas gramaticales. Es de mucha utilidad que los PLN se puedan ampliar fácilmente, pues así permiten su sencilla adaptación a todo tipo de usos particulares.

Problemas que presentan los PLN

Del conjunto de problemas que afectan a los procesadores de lenguaje natural vamos a citar los más importantes. Un conjunto de problemas surge de la gramática del lenguaje. Lo ideal sería un procesador que tuviera toda la gramática incluida, pero es

obvio que, por el momento, esto excede la capacidad de los ordenadores, al existir un número muy importante de reglas en una sintaxis completa. La solución apunta hacia dos vías: desarrollar sólo un subconjunto de la lengua, aplicando unas cuantas reglas sintácticas, o bien obviar la falta de una gramática completa construyendo un sistema que tenga en cuenta primeramente consideraciones de tipo semántico.

Una dificultad tan importante como la anterior son las preguntas gramaticalmente mal construidas. Ciertos investigadores se inclinan por construir procesadores que se anticipen al error incluyéndolo en las reglas sintácticas, mientras que otros, haciendo menos hincapié en consideraciones sintácticas, logran sistemas menos sensibles a dichos errores.

Otro problema a considerar, por su amplia utilización, es el uso de pronombres. Al aparecer uno, el analizador debe resolverlo antes de poder enviar la pregunta al sistema. Veamos con un ejemplo la importancia de éste factor.

* ¿Tienen todas las chicas moto?
Sí
* ¿Cuántas tienen más de 16 años?

El usuario tiene en mente que la edad de conducir una moto ronda los 16 años, por lo cual la segunda pregunta alude a las chicas. El ordenador, por su parte, puede no ser partícipe de dicho conocimiento y no saber cuál de los nombres —Chica, —Moto concuerda con la pregunta.

Estos y otros problemas dan una idea de la complejidad de los procesadores de lenguaje natural y del largo camino de investigación que queda por recorrer. Si se piensa que la solución pasa porque el usuario hable mejor se desvirtúa el propósito de estos interfaces, que es permitir al profano comunicarse en su propio idioma. Todo intento de poner restricciones es desvirtuar su finalidad.

Como resumen podemos decir que los procesadores de lenguaje natural no llegarán al mercado hasta dentro de unos años debido fundamentalmente a tres factores:

- Se requieren grandes ordenadores para su implantación. No sólo para contener al PLN, sino también el sistema al que va asociado debe ser de cierta importancia y capacidad.
- Tipificar todo el lenguaje es una ardua tarea, por lo que, de momento, se utilizarán lenguajes parciales.
- El lenguaje humano es poco apto para transmitir información clara, concisa y precisa.

Arquitectura de un procesador de lenguaje natural

Un procesador de lenguaje natural es uno de los módulos que componen un sistema de recuperación de información. Su misión es analizar, tanto semántica como sintácticamente, las preguntas del usuario y crear una estructura formal que luego se traduzca al lenguaje propio del sistema asociado.

El proceso que sigue una pregunta se puede resumir mediante un diagrama de bloques (Fig. 1); el procesador de lenguaje natural abarca únicamente los tres primeros módulos.



Figura 1.—Flujo de trabajo de un procesador de lenguaje natural. Partiendo de una pregunta se crea una estructura única comprensible por el sistema asociado.

Casi todos los productos que hoy en día se comercializan en el mercado presentan esta distribución. El intérprete francés SAPHIR, como se puede apreciar en la figura 2, tiene una gran similitud con el esquema típico de estos sistemas. En él se pueden apreciar los módulos de corrección de errores morfológicos y léxicos, así como el que realiza la función de concretar, estableciendo un diálogo con el usuario, las preguntas ambiguas o imprecisas.

Una vez visto de forma general dónde se enclava un PLN, el siguiente paso es describir en mayor detalle los distintos módulos que componen dicho sistema (Analizador, Diccionario y Direccionador) y su implantación en diversos programas.

EL ANALIZADOR

El analizador es el programa que estudia la gramática de la oración. Dicha gramática definirá la función de cada palabra dentro de la frase. La clave del éxito o del fracaso de un procesador de lenguaje natural es la representación de dicho lenguaje.

Entre los distintos tipos de representaciones las más comúnmente usadas son los árboles funcionales y las estructuras temáticas, apoyándose ambas en la forma en que una idea se expresa en palabras y frases. En la primera el analizador (o parser) crea

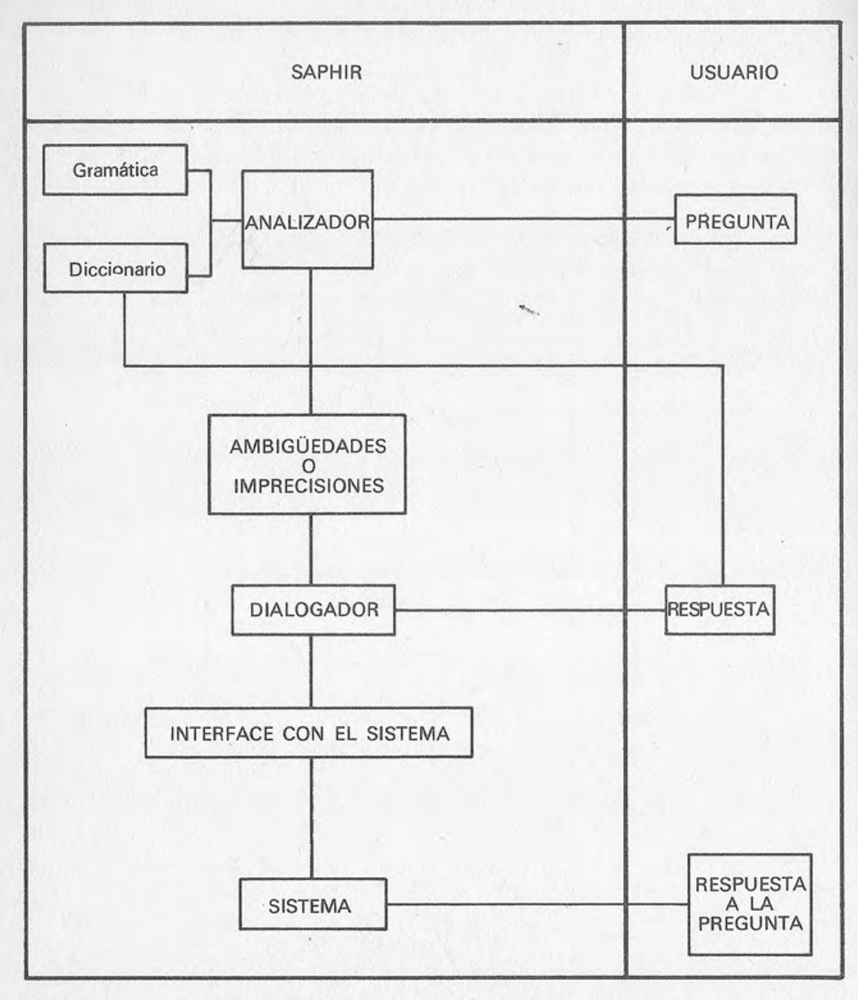


Figura 2.—Arquitectura real de un PLN, el SAPHIR, desarrollado en Francia como interface para una base de datos inteligente.

los árboles a partir de las reglas sintácticas de construcción de oraciones, mientras que el procesador de estructuras determina cómo los distintos sintagmas nominales se relacionan con el verbo.

Si se considera la frase "Begoña compró unas manzanas rojas en el mercado", el árbol funcional se compondría (tal y como se aprecia en la figura 3) de un sintagma nominal, más el sintagma

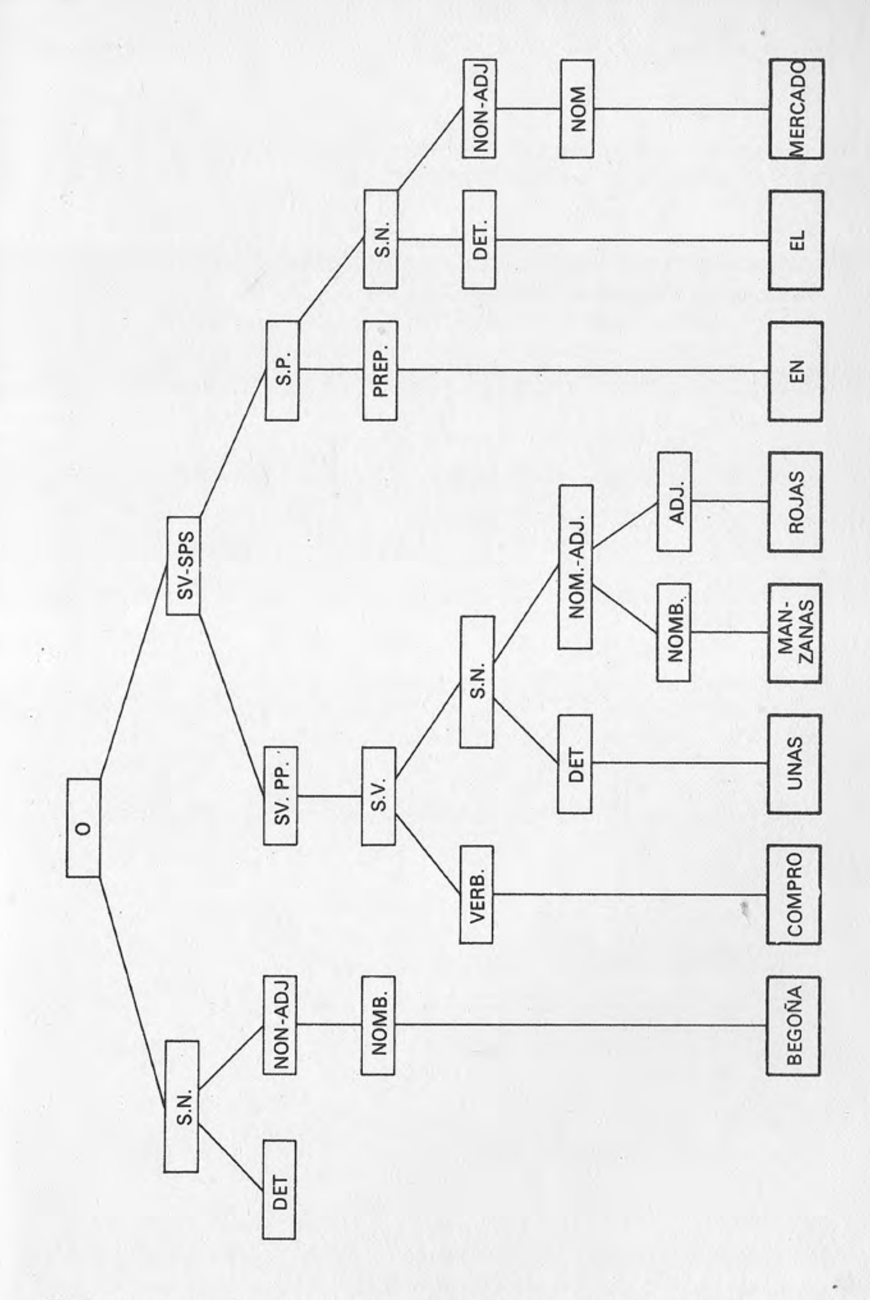


Figura 3.—Arbol funcional resultado de un análisis sintáctico mediante una gramática de libre contexto.

verbal, compuesto a su vez del verbo con su sintagma nominal, y la frase preposicional con su sintagma nominal correspondiente.

Por otra parte, el procesador de estructuras temáticas indicaría que el acto es "comprar", el agente "Begoña", el objeto temático "manzanas rojas" y el lugar o procedencia "en el mercado" (Fig. 4).

Como vemos, los métodos son completamente diferentes: en los árboles el fin del analizador es determinar la función de cada palabra dentro de la oración, mientras que en las estructuras temáticas es determinar cómo se relaciona cada sintagma nominal con el verbo. Son dos teorías que en algunos casos se pueden complementar, *la sintaxis* y *la semántica*.

Dentro de los árboles funcionales se han definido cuatro métodos de construcción de los mismos: la gramática de libre contexto, las redes de transición, las redes de transición aumentadas (ATN) y el analizador WASP (Wait-and-see), siendo cada uno de ellos más complejo que el anterior.

Por el contrario, en las estructuras temáticas sólo existe un método: averiguar el papel que los sintagmas nominales juegan dentro de la oración. Uno de ellos se identificará con el sujeto que rea-

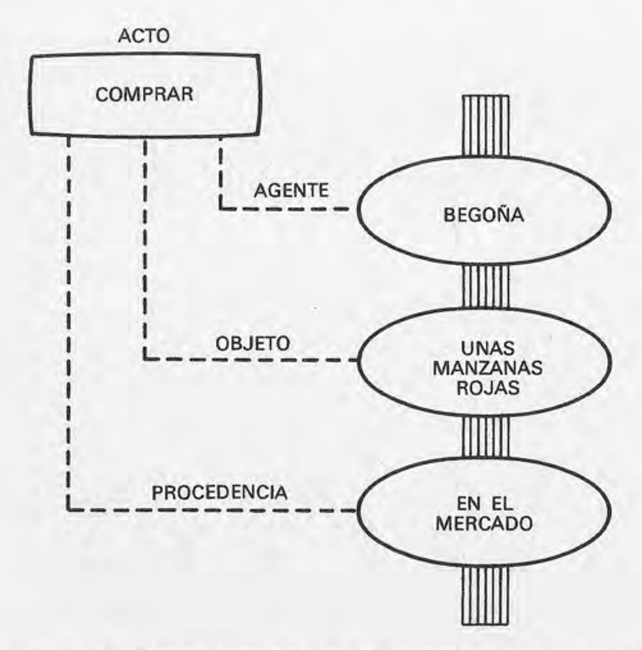


Figura 4.—Estructura temática de la oración "Begoña compró unas manzanas rojas en el mercado". El análisis es semántico.

liza la acción, otro con lo que se realiza (objeto directo) y otros con diversas circunstancias. El número de papeles temáticos varía según el tema o la persona que construya el sistema pero se pueden destacar las siguientes:

— Sujeto o Agente	Begoña se comió un pastel.
— Objeto temático	Nacho rompió la radio.
— Instrumento	Nacho rompió la radio con *un martillo*.
— Lugar	Carlos estudia *en casa*
— Beneficiario	Alicia tomó un plátano *para Antonio*.
— Destino	Los alumnos se fueron *a Madrid*.

EL DICCIONARIO

La función fundamental de un diccionario, aparte de almacenar un conjunto de palabras, es representar las relaciones entre las mismas. Una palabra puede representar una relación, un atributo o un valor. Igual que el analizador debe contener categorías gramaticales y, por otra parte, un gran número de significados.

El diccionario puede estar estructurado en forma de redes, donde cada nudo es una palabra y el arco entre dos nudos la relación entre los mismos, o bien en forma de campos jerárquicos.

Una característica que debe incorporar el diccionario es la posibilidad de aceptar nuevas palabras que el usuario va introduciendo en la conversación; por ejemplo:

(usuario)	— El profesor da clases de LATIN.
(ordenador)	* No entiendo la palabra LATIN
	1) Es un sitio.
	2) Es un idioma.

La próxima vez el diccionario identificará la palabra latín correctamente.

EL DIRECCIONADOR

La representación interna de las preguntas en lenguaje natural es uno de los aspectos claves en un procesador. De todas las posibles representaciones se debe optar por la que más fácilmente se ajuste al idioma utilizado. No deben existir, al contrario que en el lenguaje natural, inexactitudes de ningún tipo: todos los predicados en la representación interna deben ser precisos. Una ambigüedad puede producir errores posteriores. Asimismo, la estructura funcional (es decir, cómo están ordenados funcionalmente los distintos objetos) debe quedar explicitada en el predicado.

Procesadores de alto nivel

Los procesadores de alto nivel conectan con sistemas de recuperación de información no tradicionales; su campo de acción son los sistemas expertos, con los que forman un conjunto de prestaciones muy apreciables. Para dar una idea de la capacidad de estos procesadores veamos cómo es capaz de generalizar ideas y catalogar conceptos un sistema denominado "Nano Klaus":

Un submarino es un barco.
* ¿Está diciendo que todo submarino es un barco?
Sí
* O .K. Ahora he asimilado lo que es un submarino.
Un destructor es un barco.
* Está diciendo que todo destructor es un barco.
* ¿Es el submarino una clase de destructor?
No.
* ¿Está el submarino compuesto de diferentes elementos que el destructor?
Sí.
* ¿Componen el submarino y el destructor todos los barcos?
No.
.
.
.

A partir de esta conversación el sistema clasificará a los submarinos y a los destructores correctamente dentro del sistema.

Los sistemas de alto nivel no están en la actualidad comercializados y sólo existen unos pocos prototipos en experimentación.

CAPITULO VI

LENGUAJES DE PROGRAMACION Y HERRAMIENTAS DE I.A.

Una vez que se han estudiado los métodos de representación del conocimiento y las técnicas de control se plantea la eterna pregunta: ¿Cómo se consigue crear un sistema experto? La respuesta, en su origen, es clásica: puesto que un sistema experto es un programa de ordenador, para realizarlo habrá que utilizar un lenguaje de programación.

Si se responde un poco más en profundidad la respuesta hay que realizar algunos matices. Obviamente hace falta un lenguaje de programación, pero habrá unos lenguajes que se adapten mejor a nuestras necesidades que otros; incluso se pueden utilizar herramientas más evolucionadas que los lenguajes clásicos, aunque estén basados en ellos.

Basándose en los diversos niveles de software se han creado y diferenciado tres herramientas distintas que posteriormente se analizarán: los lenguajes de propósito general, los sistemas esqueletos y los sistemas conchas. Cada uno de ellos va aumentando en sencillez de utilización para el usuario, pero, a su vez, va perdiendo generalidad en sus aplicaciones.

Niveles de Software

La figura 1 muestra los niveles en que se divide el software que se necesita para resolver un problema mediante el hardware de un ordenador. En la parte superior de la figura se refleja el problema al que se enfrenta el experto en la materia, una parte del

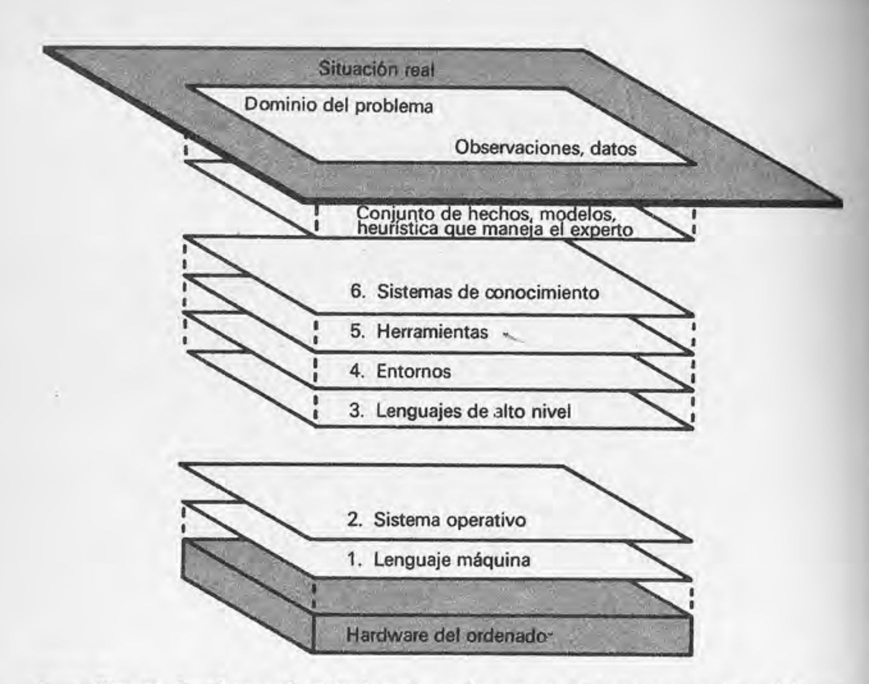


Figura 1.—Los seis niveles de software posibles entre el problema real y el hardware del ordenador.

mundo real; cuando se desea resolver dicho problema todo lo que observamos es cómo el experto maneja los datos de entrada y a partir de ellos llega a unas conclusiones. Lo que no se aprecia son los modelos mentales, las reglas de decisión y las estrategias que usa el experto cuando decide qué hacer en una determinada situación particular. Este tipo de conocimiento es el siguiente escalón en la resolución de cualquier situación y es realmente lo más imprescindible.

Por otra parte, independientemente del software que se utilice, todos los sistemas expertos dependen en el fondo del hardware del ordenador. La parte física del mismo es, al fin y al cabo, la encargada de manejar en último término toda la información en forma de códigos binarios, una larga serie de ceros y unos que el ordenador convertirá en respuestas físicas discretas. Este es el nivel inferior del software denominado lenguaje máquina.

El nivel inmediatamente superior es el programa que dirige y direcciona las operaciones fundamentales del ordenador: el sistema operativo; se encarga de controlar las sentencias del lenguaje máquina. Suele estar escrito en lenguaje máquina o, recientemente, impreso en hardware dentro de un chip.

Introduciéndonos ya en lo que es propiamente la programación, casi toda ella se realiza en los denominados lenguajes de alto nivel. Los más conocidos son Basic, Cobol, Pascal, Fortran, Forth C, etc. Sin embargo, los programadores en inteligencia artificial utilizan fundamentalmente dos lenguajes: LIPS y PROLOG. LISP consiste en un conjunto de instrucciones que facilitan la creación de programas que manejan listas, mientras que el PROLOG facilita el trabajo con expresiones lógicas. Ambos lenguajes son muy útiles por su carácter simbólico, mientras que los otros trabajan mejor con cálculos numéricos.

Justo por encima de los lenguajes de alto nivel se encuentran los *entornos de programación*. Dichos entornos suelen estar asociados con un determinado lenguaje y contienen un conjunto de instrucciones escritas en dicho lenguaje, muy útiles para ciertas tareas de programación. Para los conocedores de lenguajes estructurados se pueden comparar con las subrutinas.

Las herramientas son el quinto nivel en la figura 1. Estas herramientas se han creado para ayudar al rápido desarrollo de los sistemas expertos. Los aspectos más importantes que presentan son las estrategias de control, representación del conocimiento e inferencia comunes a la mayoría de los sistemas expertos. La razón de una herramienta de este tipo es similar a las de una carpintería: en vez de crear una nueva para cada tipo de mueble, se utilizan aquellas que nos han sido útiles en anteriores ocasiones. Por supuesto, se debe usar la herramienta más apropiada para cada situación; cada una de ellas está especialmente diseñada para un trabajo en particular.

Cuando se combina una herramienta con el conocimiento sobre un tema específico, el resultado es un sistema experto basado en el conocimiento (nivel 6). Dicho sistema, si está bien desarrollado, presenta la misma capacitación que el experto en dicho tema.

De los lenguajes a los sistemas concha

En general los lenguajes son más flexibles, pero más difíciles de usar en la creación y puesta a punto de un prototipo rápidamente. Sólo cuando se necesita una aplicación muy particular o cuando el programador está muy bien preparado se construyen sistemas basados en LISP o PROLOG. Los sistemas conchas son menos flexibles, ya que llevan incorporado un particular sistema de control. Como consecuencia, si se tiene una herramienta apropiada a nuestro problema, el desarrollo es muy rápido, e incluso personas con muy poca experiencia pueden crear pequeños pero muy útiles sistemas basados en el conocimiento.

En la figura 2 se presenta una vista general de los sistemas y lenguajes más conocidos. Los sistemas esqueleto o entornos se sitúan a mitad de camino entre los lenguajes y los sistemas concha, y sus características son una mezcla de flexibilidad y facilidad de utilización.

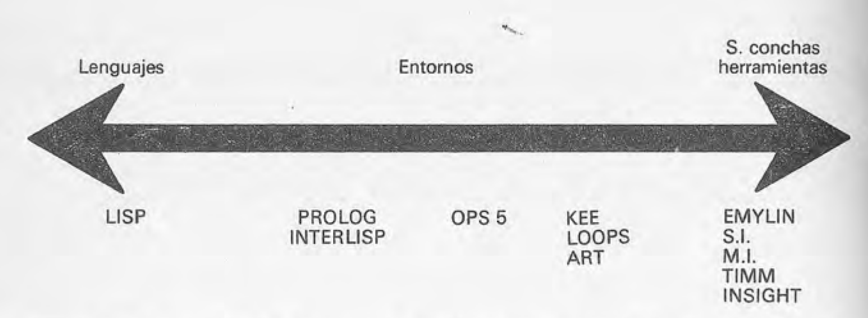


Figura 2.—De los lenguajes de propósito general (LISP, PROLOG) a las herramientas más específicas. EMYCIN, S1, ...

LISP es un lenguaje puro, mientras que PROLOG o INTERLISP están más cerca de los entornos. OPS 5, desarrollado por la Carnegie-Mellon University, sería propiamente un entorno; contiene un determinado sistema de control de las reglas, pero es bastante flexible. KEE es una herramienta o sistema concha, pero híbrida, es decir: puede optar por diversos controles y formas de representar el conocimiento; sin embargo, es muy difícil de utilizar. Por último EMYCIN, S1, etc., son muy fáciles de usar, pero se ajustan a un tipo específico de sistemas expertos.

La figura 3 introduce una nueva complejidad en el análisis. Los lenguajes de programación en Inteligencia Artificial suelen ser LISP o PROLOG, pero es cierto que un sistema experto se puede escribir en Fortran o Pascal, por ejemplo. ¿Cuáles son las ventajas y los inconvenientes de su utilización? Los lenguajes de Inteligencia Artifical tienen unas características que les posibilitan el manejo de símbolos, mientras que los lenguajes convencionales trabajan mejor con números. Es mucho más conveniente, por tanto, tener herramientas o entornos programados internamente en lenguajes LISP o PROLOG que en lenguajes clásicos.

La más clara desventaja para los "especializados" se presenta en que son menos conocidos y están menos extendidos que los

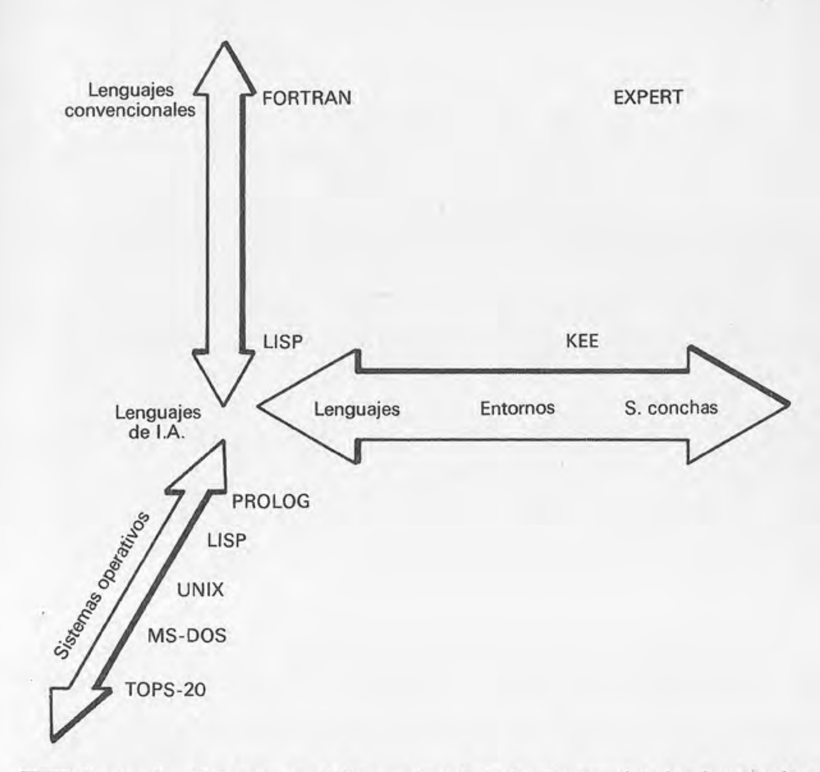


Figura 3.—Estructura tridimensional de los lenguajes, las herramientas y los sistemas operativos en que trabajan.

clásicos lenguajes de cuarta generación: mientras que casi todos los ordenadores tienen implantados FORTRAN y PASCAL, pocos de ellos tienen LISP o PROLOG. Muchos de los sistemas concha están, por tanto, escritos en PASCAL, lo que facilita su implantación en toda clase de ordenadores, aunque al final sean menos eficaces.

Otro problema radica en el modo ineficaz en que los sistemas operativos convencionales traducen el LISP a lenguaje máquina, lo que provoca la lentitud de trabajo de muchos sistemas expertos. El futuro más próximo es la construcción de máquinas cuyo sistema operativo esté escrito en LISP.

El estudio de los lenguajes de programación no es tarea de este libro y por su importancia e interés será objeto de otro volumen de la B.B.I., pero sí queremos ofrecer una pequeña introducción a las características más sobresalientes de los más importantes.

LISP

Se puede decir que el lenguaje LISP es el único utilizado por los americanos dentro de la inteligencia artificial. LISP proviene de List Processing Languaje; fue creado por John McCarthy en 1958 y es, por lo tanto el más antiguo de los lenguajes en uso después del FORTRAN. Su autor resumió en 1978 sus características más importantes:

- Trabaja con expresiones simbólicas más que con números; esto es, los bits de la memoria y los registros pueden ser ocupados por símbolos arbitrarios y no sólo por operadores aritméticos.
- Procesamientos de listas; representación de los datos en listas estructuradas dentro de la memoria
- Estructura de control basada en la composición de funciones básicas para formar otras más complejas.
- Utilización de la Recursividad como una forma de escribir procesos y resolver problemas.
- Representación interna de los programas escritos en LISP como un conjunto estructurado de listas.
- La función EVAL, escrita a su vez en LISP, sirve como un intérprete del mismo y una definición formal del lenguaje.

La conclusión más importante que se deduce es que no existen diferencias esenciales entre los datos y las sentencias del programa, es decir: un programa LISP puede usar unos datos que, a su vez, son otro programa LISP. Esta característica se traduce en la facilidad de modificar y ampliar un programa LISP. En consecuencia, el sistema experto asociado es capaz, por una parte, de modificarse a voluntad y, por otra, de diferenciar claramente el control y las reglas, aunque en el fondo ambas cosas sean programas LISP.

Existen muy pocas funciones básicas en LISP; todas las demás se desarrollan a partir de éstas. Partiendo de la unidad fundamental, la lista (que a su vez se compone de unidades indivisibles denominadas átomos) se crea la red del programa. Dicha red se construye mediante los nudos CONS. Cada átomo tiene asociado una lista de propiedades que contiene información sobre el átomo: nombre, valor y cualquier otra propiedad que se desee. Los nudos CONS son una estructura de datos dinámica que se compone de dos campos (Fig. 4), cada uno de los cuales apunta hacia otro dato. Estos encadenamientos son fácilmente modificables.

El manejo de estas listas se realiza mediante las siguientes funciones:

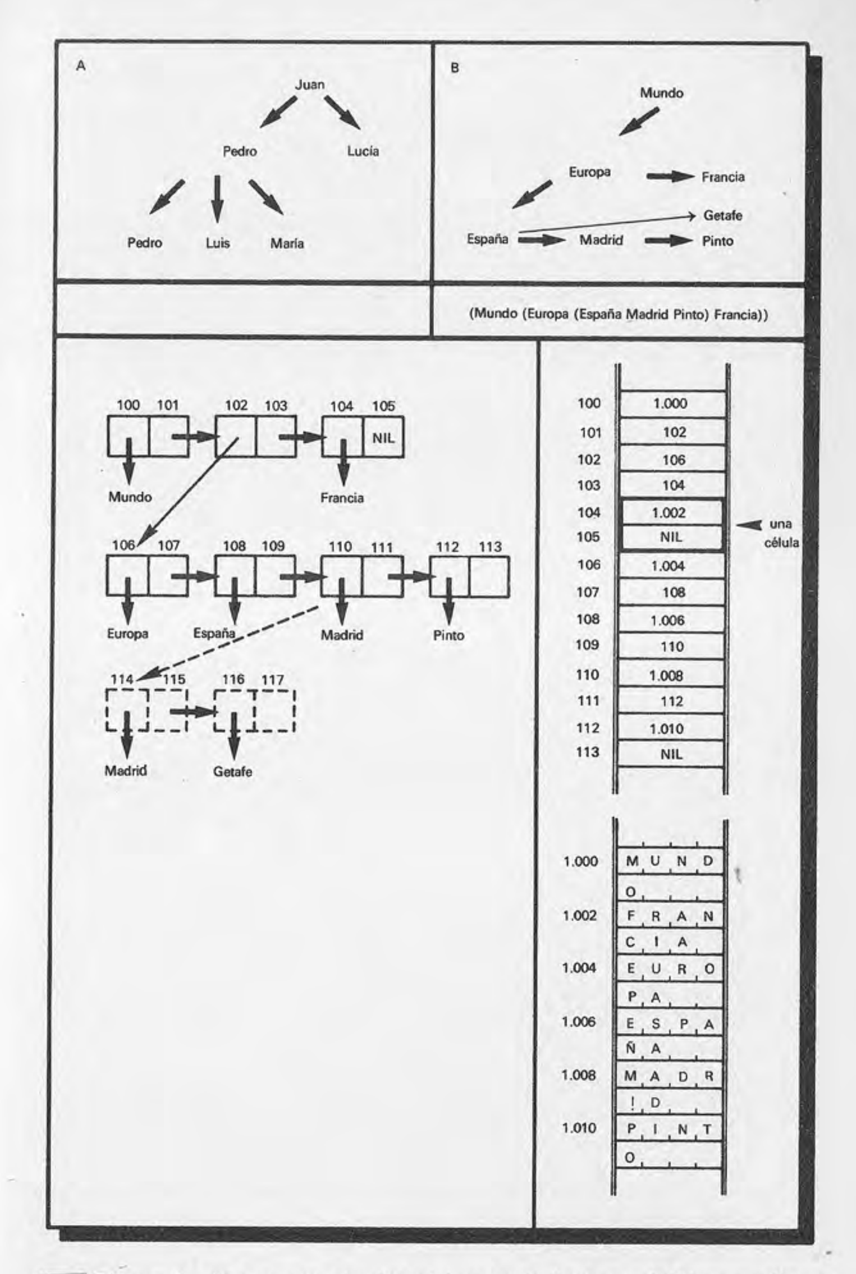


Figura 4.—La representación del conocimiento en LISP se suele realizar mediante árboles o grafos de relaciones. Dichos grafos se representan mediante listas, fácilmente codificables en memoria.

- CAR Proporciona el valor del primer elemento de una lista:

  CAR (Come Juan Patatas) = "Come"

- CDR Da el valor de la lista sin el primer elemento:

  CDR (Come Juan Patatas) = (Juan Patatas)

- CONS Junta dos expresiones para formar una lista:

CONS	COME átomo	(JUAN) = (COME JUAN) lista
CONS	LUIS átomo	( ) = (LUIS) lista vacía

- ATOM Analiza si una estructura es lista o átomo; si es lista devuelve el valor (False)
- EQUAL Proporciona el valor (True) si las dos expresiones son iguales.

La programación en LISP se lleva a cabo construyendo funciones muy sencillas que realizan unas determinadas tareas y luego agrupando las mismas para completar el trabajo global.

El LISP no ha evolucionado mucho a lo largo de los últimos años debido a su carácter experimental que ha tenido hasta ahora, y a sus características propias; sin embargo, en el cuadro de la figura 5 se pueden apreciar las tendencias que ha seguido.

PROLOG

PRLOG, abreviatura de PROgraming languaje for LOGic, fue desarrollado inicialmente por Colmeraver y P. Roussel en 1972 en la Universidad de Marsella.

PROLOG es un lenguaje de programación que implementa una versión simplificada del cálculo de predicados y utiliza un lenguaje lógico. Ha sido últimamente muy popularizado debido a su inclusión dentro del programa sobre quinta generación de ordenadores que llevan a cabo los japoneses; también se está utilizando en los programas de Inteligencia Artificial que están siendo desarrollados por ingleses y franceses.

Para programar en PROLOG se siguen los siguientes pasos:

1º — Especificar algunos hechos y relaciones.

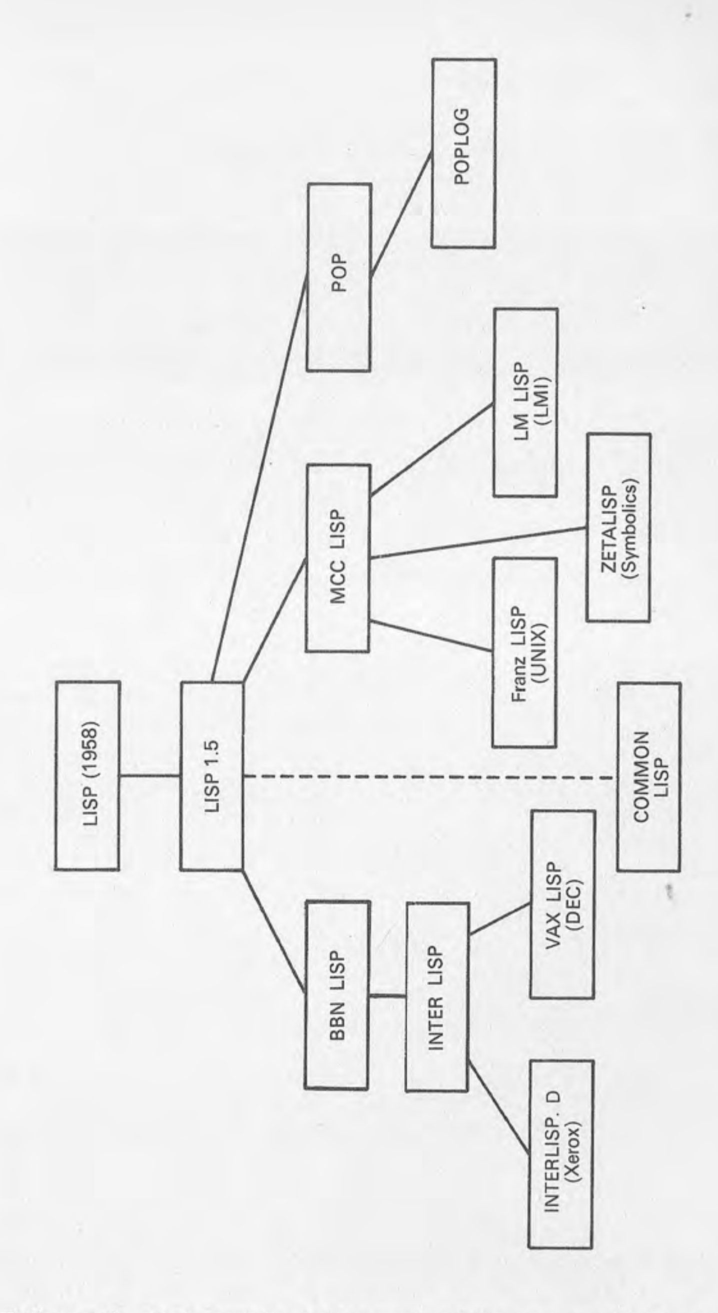


Figura 5.—El desarrollo del lenguaje LISP se ha producido durante los últimos años, pero las diferencias no son demasiado grandes.

2º — Especificar unas reglas sobre dichos hechos y relaciones.
3.º — Hacer preguntas sobre los mismos.

Por ejemplo:

Ama (Carlos Ana)
Ama (Pepe Julia)
Ama (Gustavo Merche)

Ahora, si preguntamos:

? Ama (Pepe Julia)

PROLOG responderá

SI.

En este ejemplo tan trivial, "Ama" es un predicado que indica una relación entre Carlos y Ana.

Aunque PROLOG no incorpora toda la lógica que se necesita en el cálculo de predicados, su sintaxis es mucho menos compleja que la mayoría de los lenguajes normales de potencia comparable. Un lenguaje no puede ser estrictamente lógico: necesita algunos códigos básicos que controlen los aspectos de procedimiento de las operaciones; esto, y en el grado mínimo, es lo que incorpora Prolog. De esta forma, en PROLOG se dice al ordenador en un lenguaje declarativo "qué" tiene que hacer y él se encargará de "cómo" hacerlo. Este es el aspecto más sobresaliente que incorpora PROLOG y lo que le diferencia de los demás lenguajes de programación.

POPLOG

Es una combinación de LISP, PROLOG y POP-11 introducidos en un mismo paquete; cuando se compila es más rápido que LISP o PROLOG incorporando además las ventajas de ambos sistemas. Es el sistema que está desarrollando IBM para sus aplicaciones internas.

Sistemas esqueleto o entornos de programación

Sobre un sistema experto ya construido y bien probado se recoge el mayor número de elementos compatibles con otros, formando así un sistema esqueleto. Por ejemplo, la forma de representar el conocimiento, el control de las reglas, la inferencia, etc., representan el esqueleto del antiguo sistema experto.

Cuando los creadores de un sistema experto pionero sobre diagnóstico médico (MYCIN) acabaron su trabajo se dieron cuenta que lo podían separar en dos partes bien diferenciadas: la base de conocimientos, donde se contenía toda la información sobre enfermedades infecciosas y que era específico de esta aplicación, y el motor de inferencia, un control "backward chaining" de propósito general que bien podía ser utilizado por otras personas para crear otro sistema experto de parecidas características, pero sobre un tema completamente distinto (Fig. 6).

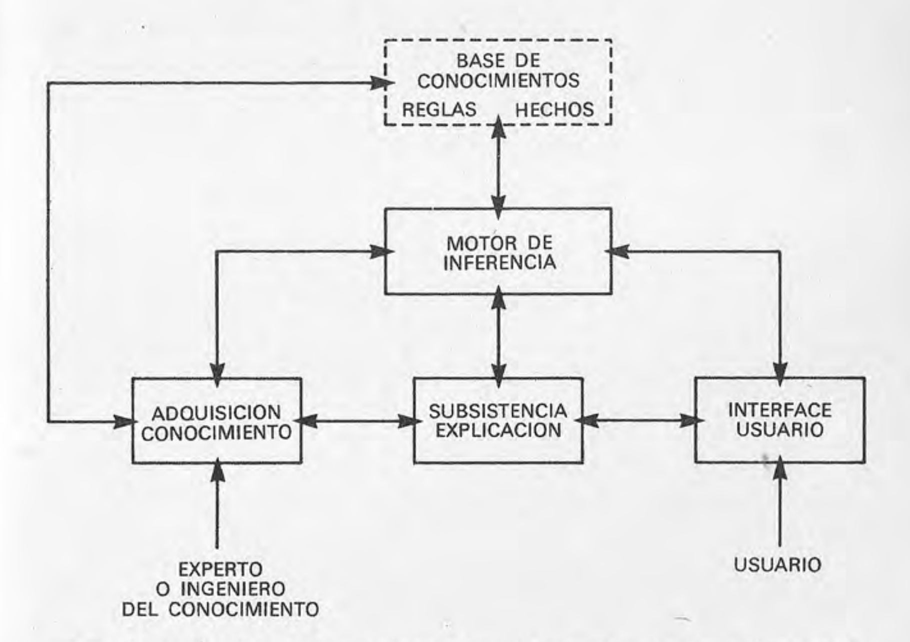


Figura 6.—Arquitectura de un sistema esqueleto: incluye el motor de inferencia y los subsistemas de adquisición, explicación e interface con el usuario.

EMYCIN es una parte de MYCIN que recoge todo excepto la base de conocimientos, pero que contiene todo lo necesario para razonar sobre una base de conocimientos y responder a las consultas del usuario. No es, por tanto, un lenguaje de propósito general que se ajuste completamente a los requerimientos del sis-

tema experto, sino que al utilizar reglas O-A-V, usar la lógica modus ponens y el motor de inferencia "backward chaining" sólo se puede adaptar a un tipo concreto de sistemas expertos, los de diagnosis fundamentalmente.

Casi todos los sistemas esqueletos o entornos de programación han ido apareciendo asociados a un sistema experto concreto, al cual se le ha suprimido la parte de la base de conocimientos. La figura 7 representa la evolución de diversos sistemas en entornos de programación y su desarrollo a lo largo de los últimos años.

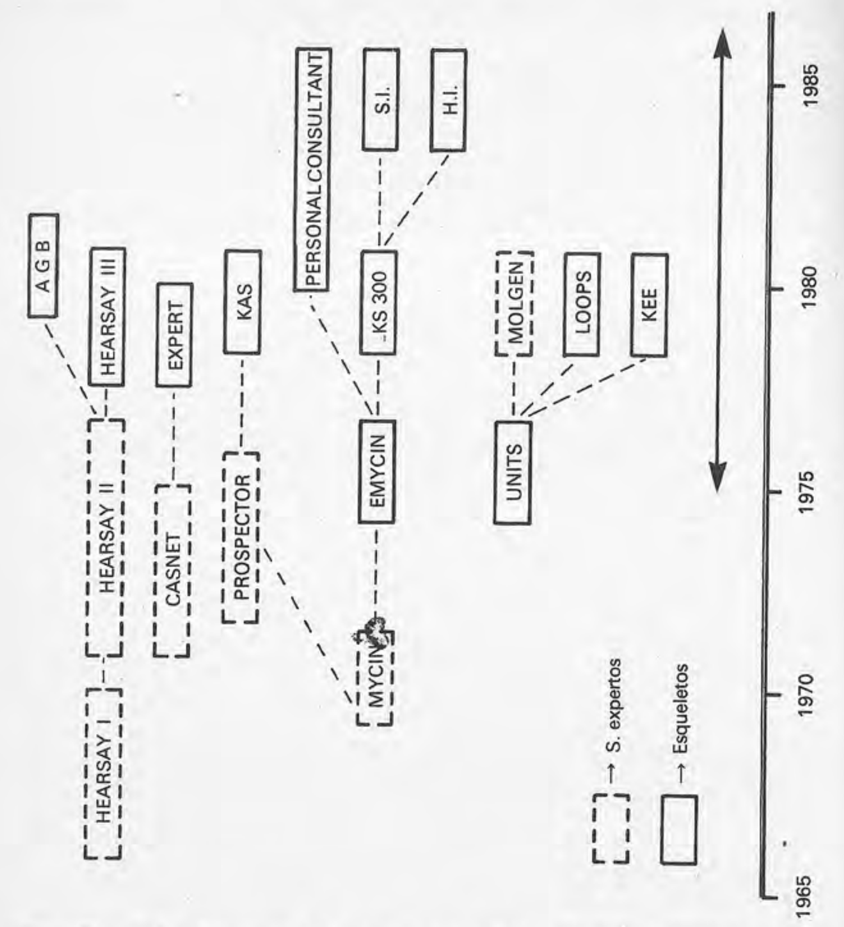


Figura 7.—Desarrollo a lo largo del tiempo de los sistemas expertos y los sistemas esqueletos a que han dado lugar.

Como se ha observado, el utilizar parte de algo ya creado es una buena solución, pero sin embargo, no está exenta de ciertos problemas, sobre todo si se quiere aplicar en tareas distintas a las concebidas para el sistema original. Este sería el caso de utilizar EMYCIN (diagnosis fundamentalmente) para crear un sistema experto de ayuda a la planificación (EGERIA).

Los principales problemas son:

— El armazón del antiguo sistema puede resultar inapropiado para la nueva tarea a realizar.

— La estructura de control puede no encajar perfectamente en la forma de realizar inferencias que se desea.

— El lenguaje antiguo puede resultar inapropiado.

— Puede existir en el sistema un conocimiento previo sobre el control que no se necesite.

Sistemas conchas o herramientas

Son sistemas desarrollados a partir de lenguajes de propósito general que proporcionan mayor potencia para la creación de los sistemas expertos. Incorporan muchas facilidades sobre inferencia y control y agilizan enormemente la creación del sistema.

Los expertos en Inteligencia Artificial, o Ingenieros del Conocimiento, estudiando el tipo de problemas a los que más fecuentemente se enfrentan los profesionales de diversos campos han identificado tres estrategias de resolución, bien definidas y claramente diferenciadas. Son la diagnosis o prescripción, la planificación y el diseño. En la figura 8 se pueden observar las características de cada estrategia y el tipo de problemas a los que se enfrenta. Una vez que los pioneros definieron estos tres tipos, los sistemas conchas que se han construido o se están construyendo se diseñan fundamentalmente para una de estas aplicaciones o, a lo sumo, para dos de ellas y sólo permiten el desarrollo de sistemas expertos cuya tarea se enclave dentro de los sistemas resolubles por dichas estrategias.

Ciertas técnicas de representación del conocimiento (de inferencia o las estrategias de control) son en muchos casos específicas de una aplicación; por ejemplo, para la planificación, lo más normal es utilizar "forward chainning" (encadenamiento hacia adelante) en vez de "backward chainning", pero, sin embargo, otras, como por ejemplo los factores de certidumbre (centainty factors) pueden ser utilizadas en más de una aplicación, estando, por tanto, incorporadas en la mayoría de los sistemas conchas. En general problemas de un tipo particular llevan asociadas herramientas

Tipo de problema:	Diagnosis/Prescripción	Planificación	Diseño
Ejemplo	Diagnóstico de enfermedades infecciosas y prescripción de medicamentos	Planificar un proyecto. Configurar un ordenador	Diseño de un circuito impreso
Entrada	Condiciones, respuestas a preguntas	Fines, limitaciones, medios	Fines, limitaciones
Conocimiento	Uso de reglas y cálculos sencillos	Heurística, modelos, restricciones	Heurística, modelos, restricciones
Salida	Soluciones heurísticas	Planificación heurística	Descubrimientos heurísticos

Figura 8.—Tipos normalizados de problemas: la diagnosis, la planificación y el diseño.

construidas con una determinada forma de representación, inferencia y control.

Los problemas o paradigmas más fáciles de resolver son los de Diagnosis y prescripción y en la actualidad la mayor parte de los sistemas concha comerciales (Fig. 9) están preparados para estas aplicaciones; únicamente los más complejos son capaces de atacar los problemas de planificación y, hoy por hoy, herramientas comerciales que ayuden a crear sistemas inteligentes en diseño no existen.

ENTORNO/ HERRAMIENTA	Diagnosis/ Prescripción	Planificación	Diseño
EMYCIN	●	○	○
ES/P ADVISOR	●	○	○
Expert. Ease	●	○	○
INSIGHT	●	○	○
M.I.	●	○	○
Personal Consultant	●	○	○
EXPERT	●	○	○
KES	●	○	○
OPS 5	●	●	○
S.I.	●	○	○
TIMM	●	○	○
ART	●	○	○
KEE	●	●	○
LOOPS	●	○	○





Figura 9.—Herramientas comerciales más usuales y tipos de problemas a los que se enfrentan.

Comercialmente los sistemas concha se dividen en tres categorías:

- Herramientas para pequeños sistemas. Pueden trabajar sobre ordenadores personales. Son diseñadas para facilitar el desarrollo de sistemas que contengan menos de 400 reglas.
- Herramientas para grandes sistemas específicos. Pueden trabajar sobre máquinas LISP (ordenadores especiales) o en grandes ordenadores y están diseñadas para crear sistemas expertos que contengan entre 100 y varios miles de reglas, aunque estén restringidos hacia la resolución de un único tipo de problemas.

	CONOCIMIENTOS									
	HECHOS			REGLAS LOGICAS					INCERTIDUMBRES	
● Función presente ◐ Presencia restr. ○ No presente, pero program.	Pareja A-V	Terna O-A-U	FRAMES	Reglas IF-THEN	Reglas variables	Ejemplos	Múltiple-objects	Inheritance	Coeficiente de certeza	Probabilidad
Pequeñas herramientas										
H.I.	●	○		●	●		○	◐	●	○
Personal Consultant		●		●			●	◐	●	
Grandes sistemas restringidos										
S.I.		●		●			●	◐	●	
KES	●			●			●	●		●
Grandes sistemas										
KEE	○	○	◐	◐			◐	◐	○	○
LOOPS	○	○	◐	◐			◐	◐	○	○

Figura 10.a).—Características de los sistemas concha más comunes en función de la representación del conocimiento.

- Herramientas para grandes sistemas. Pueden trabajar sobre máquinas LISP o en grandes ordenadores y están diseñadas para crear sistemas que contengan entre 500 y varios miles de reglas; pueden incluir las características necesarias para la resolución de distintos tipos de problemas.

En la figura 10 se intenta dar una vista general de las diferencias que presentan varios sistemas conchas pertenecientes a las tres categorías anteriores; se puede apreciar cómo al ir aumentando la potencia aumentan las posibilidades de poder utilizar el conjunto de representación del conocimiento, inferencia y control que mejor se adapta al problema.

	INFERENCIA										
	Generación de nuevos hechos				Estrategias de control						
● Función presente ◐ Presencia restringida ○ No presente, pero programable											
Pequeñas herramientas											
M.I.		●			●	◐	●				
Personal Consultant		●			●	◐	●				●
Grandes sistemas restringidos											
S.I.		●			●		●		●		●
KES					◐				●		
Grandes sistemas											
KES		○			○	○	○	○	○	○	○
LOOPS		○			○	○	○	○	○	○	○

Figura 10.b).—Características de los sistemas concha más comunes en función de los motores de inferencia.

CAPITULO VII

CREACION DE UN SISTEMA EXPERTO

A través de este capítulo intentaremos dar una idea del proceso real que se sigue en la construcción de un sistema experto pequeño, basado en la utilización tanto de las técnicas de inteligencia artificial como de las herramientas que se han puesto al servicio de los ingenieros del conocimiento.

La aplicación y el tipo de conocimiento se deben ajustar a los modelos ya estudiados de forma que su análisis sea menos complejo; problemas cuya resolución implique engorrosos cálculos o procesos iterativos tendrán probablemente mejor solución en la ya muy probada programación clásica, y el coste, por otra parte, será sensiblemente inferior.

Con el problema ya delimitado, los siguientes pasos se encaminarán a determinar el tipo apropiado de herramienta necesaria para atacarlos, desde la utilización de lenguajes de propósito general como LISP o PROLOG a los sistemas conchas, pasando por los sistemas esqueletos o entornos. La elección de uno u otro no sólo depende de aspectos técnicos, sino también de recursos humanos y en último término, aunque son al final los que priman, los recursos económicos.

La última etapa del desarrollo pasa forzosamente por la construcción de un modelo reducido del sistema que se quiere implantar; con dicho modelo a escala se experimentará y resolverán las primeras dificultades, sirviendo de campo de pruebas del desarrollo final. Una vez suficientemente probado se van aumentando sus prestaciones, y por tanto su complejidad, y sufrirá diversos procesos de realimentación y mejora hasta su instalación final.

Primeras consideraciones

Hasta ahora se han considerado sinónimas las acepciones "sistemas expertos" y "sistemas basados en el conocimiento" (Knowledge systems), pero en los sistemas pequeños esto no es del todo cierto. Los gráficos de la figura 1, que relacionan el número de personas que conocen un tema con lo que sabe cada una de esas personas sobre dicho tema en tres situaciones distintas, muestra cómo en algunas de dichas situaciones o no existe el experto o todos son expertos. Es decir, aquellos sistemas expertos que contengan el conocimiento de los problemas del tercer tipo de la figura 1 no modelan realmente al experto humano, sino que sólo contienen un tipo de conocimiento.

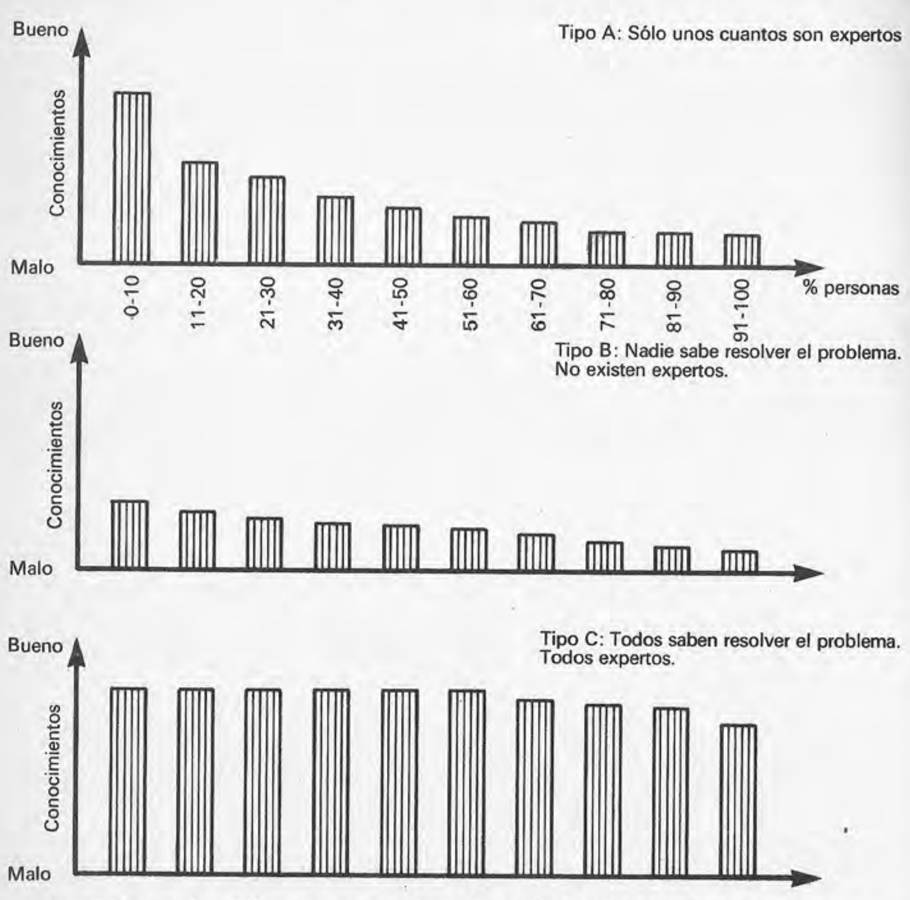


Figura 1.—Diferentes tipos de "profundidad del conocimiento" en un problema en función del número de expertos que conocen dicho problema.

Si se construye un sistema experto que ayude a decidir el tipo de los seguros, la cuantía de los mismos y los bienes que cubre, no se puede decir que el conocimiento que posee, necesario para realizar estas funciones, sea el de un experto, sino más bien se le debe denominar sistema basado en el conocimeinto. Por otra parte están, por ejemplo, los grandes programas de inteligencia artifical como DENTRAL y MYCIN que asesoran y resuelven casos médicos muy complejos y donde realmente ha sido necesario recurrir a verdaderos especialistas para poder introducir, en unos cuantos miles de reglas y hechos, todo el saber que posee la humanidad en dichos campos científicos. Estos últimos programas sí pueden ser denominados "sistemas expertos".

Los sistemas expertos "pequeños" tienen una concepción distinta a la de los grandes sistemas; las personas que utilizan estos programas no tienen por qué ser ingenieros de conocimiento, sino que normalmente, serán personas próximas al problema. En el ejemplo anterior (aquel que ayudaba en la determinación del seguro más apropiado) el supervisor del programa deberá ser el agente de seguros y, por lo tanto, todo pequeño sistema experto será eficiente y tendrá éxito si es verdaderamente sencillo en su utilización.

Como conclusión: los sistemas pequeños son normalmente sistemas basados en el conocimiento y su característica fundamental es la sencillez de programación, manejo y modificación.

Pasos a seguir

A la hora de decidirnos por el tema de un sistema experto pequeño hubiéramos podido optar por algunos muy complejos, con los cuales se pudiera vislumbrar la potencia de estos programas, pero en beneficio de una mayor claridad hemos optado por un tema de conocimiento general como es la elección del viaje en una agencia de turismo. Es en realidad un sistema basado en el conocimiento, pues no necesita un experto en su realización y resuelve problemas que cualquiera puede solucionar.

Para su construcción los ingenieros del conocimiento deben seguir los siguientes pasos:

1. Seleccionar una herramienta y tener en mente el tipo de estrategia capaz de enfrentarse al problema.
2. Identificar el problema y analizar el conocimiento que se vaya a incluir en el sistema.

3. Diseñar el sistema. Inicialmente implica construir uno con lápiz y papel mediante la utilización de diagramas de bloques, matrices y pequeños conjuntos de reglas.

4. Desarrollar con el sistema concha seleccionado un prototipo. Escribir la base de conocimientos y probarla en un número amplio de casos.

5. Ampliar y modificar el programa hasta que funcione tal y como queremos que lo haga.

6. Mantener y poner al día el sistema según se necesite.

Paso 1: Selección de la herramienta

Las herramientas (o sistemas conchas) pequeñas son menos restrictivas en sus aplicaciones que las grandes. Si un programa necesita sólo 100 reglas su programador no demanda una serie de características que exigiría si dicho programa constara de 1.000 reglas, por el simple motivo que 100 reglas son relativamente fáciles de controlar con la cabeza, mientras que si son 1.000 no hay persona que pueda trabajar con ellas.

Por esta razón todas las herramientas pequeñas se adaptan bien y fácilmente a cualquier problema, siempre que pertenezcan al tipo de diagnosis y prescripción. En este caso se pueden utilizar cualquiera de las dos herramientas ejemplo del capítulo anterior (ver fig. 10 del capítulo 6); M.1 o Personal Consultant. Ambas representan el conocimiento mediante reglas IF-THEN, y su motor de inferencia está basado en "Backward chainning" o (encadenamiento hacia atrás). La herramienta que usaremos en este ejemplo es M.1, vendida por Teknowledge Inc. Auqnue es muy cara, está bien diseñada y es muy fácil de usar. Admite toda clase de problemas que impliquen consultas del tipo diagnosis/solución o prescripción y posee unas funciones que la hacen muy apta para ser manejada por toda clase de usuarios.

Una vez decididos por M.1 y el tipo de estrategia de resolución que esto implica se debe confirmar que nuestro problema reune las siguientes características:

- El tiempo que debemos tardar en resolverlo debe estar normalmente entre los 15 y 30 minutos. Si es menos de 10 el problema es muy sencillo, y si se tarda más de 30 implica demasiado conocimiento y será muy lento.
- No debe tener que examinar diagramas o cualquier contacto físico; será tal que pueda ser normalmente resuelto a través de una conversación telefónica.
- Asimismo el proceso de resolución se podrá llevar a cabo mediante una serie de reglas y, como mucho, unos pocos cálculos. Si la solución requiere muchos cálculos es mejor la programación tradicional como, por ejemplo, una hoja de cálculo electrónica.
- El conjunto de posibles soluciones finales no debe sobrepasar unas pocas docenas. Si dichas soluciones son más de 50 habrá, probablemente, otras herramientas más aptas que las utilizadas por I.A.

El conjunto de estas normas reduce considerablemente el número de posibles temas a considerar. En el caso de la agencia de turismo, el cliente suele explicar su caso y elegir el viaje en unos 20 minutos y no se precisan muchos cálculos matemáticos, quizá unas sumas y multiplicaciones para conocer los precios exactos aplicando diversas tarifas. A través del teléfono es posible contar todo lo referente al viaje y el conjunto de soluciones básicas no sobrepasa la docena y media. Si bien es cierto que dentro de cada solución básica (por ejemplo, apartamento en la playa) existen muchas variantes, ésta es la que en principio se buscaba, y delimitará enormemente el problema.

Paso 2: Identificación y representación del conocimiento

Consiste en la explicitación de cada uno de los factores que concurren en la solución del problema, es decir, es el conjunto de datos iniciales que debe conocer el agente de turismo para poder decidir el tipo de viaje que más conviene a cada situación.

Los factores que concurren en el problema se limitarán inicialmente a seis: motivo, fecha de viaje, dinero, número de personas, lugar de residencia y edad.

- Motivo. ¿Cuál es el motivo del viaje?, es uno de los factores más destacados y puede tomar los siguientes valores discretos:

  Descanso.
  Diversión.
  Deporte.
  Turismo.

- Fecha del viaje. Temporada en la que se pretende realizar el viaje: sólo se tendrán en cuenta las estaciones

  Verano (Junio-Septiembre).
  Invierno (Diciembre-Febrero).
  Primavera, Otoño (Marzo-Mayo, Octubre-Noviembre).

- Dinero. Cantidad de dinero que se piensa gastar en el viaje de una forma global:

  Normal (inferior a 75.000 pts/persona).
  Superior (más de 75.000 pts/persona).

- Número de personas. Indica de alguna forma la preferencia o necesidad de optar por un tipo de viaje u otro; se compone de:

  Parejas.
  Familia.
  Grupos.

- Lugar de residencia. Un factor poco indicativo por una parte, pero muy exclusivo o limitativo por otra:

  Ciudad.
  Campo.
  Costas.

- Edad. La edad suele conllevar un tipo de viaje distinto. Se compone de:

  Jóvenes (<30)
  Adultos (30<edad<60)
  Ancianos (>60)

Hay que hacer notar que no todos los factores precedentes tienen la misma importancia y así deberían figurar en el sistema experto, pero por simplicidad a todos les será asignado el mismo valor. El lector, por otra parte, debe abstraerse del ejemplo particular y generalizar mediante el mismo el concepto de sistemas basados en el conocimiento.

Una vez determinados los factores, la regla general que resulta es de la forma:

Si (1) el motivo es.. *
(2) la fecha del viaje es.. *
(3) el dinero disponible.. *
(4) el número de personas es.. *
(5) el lugar de residencia.. *
(6) la edad es de.. *
Entonces se recomienda.........**
El siguiente paso será rellenar estos huecos.

Paso 3: Diseño del sistema

Inicialmente se limita el conjunto de soluciones a 10; luego, en la fase de ampliación y mejora, se podría llegar hasta un número aproximado de 40. Con estas primeras 10 soluciones se puede crear la matriz de la figura 2, donde en abcisas situamos las condiciones o factores de entrada y en ordenadas las soluciones recomendadas.

M.1, como trabaja con "backward chainning", requiere un fin del cual partir; éste es "viaje propuesto" y lo indicamos con:

Goal = viaje-propuesto

IF: OPCIONES						THEN
Motivo	Fecha-viaje	Dinero	Personas	Residenc.	Edad	
Descanso Diversión Deporte Turismo	Verano Invierno Prim.-otoño	Normal Superior	Pareja familia Grupo	Ciudad Campo Costa	Jóvenes Adultos Ancianos	SOLUCIONES
Diversión	Verano	Normal	Familia	Campo Ciudad	Adultos	1. PLAYA-APART.-ESPAÑA
Diversión	Verano	Superior	Pareja	Campo Ciudad	Adultos	2. PLAYA-HOTEL-ESPAÑA
Descanso	Verano-invierno	Normal	Pareja Grupo	Ciudad	Jóvenes Adultos	3. MONTAÑA-ESPAÑA
Deporte	Invierno	Normal	Pareja Grupo	Ciudad Costa	Jóvenes	4. ESQUI-ESPAÑA
Deporte	Invierno	Superior	Pareja	Ciudad Costa	Jóvenes	5. ESQUI-EXTRANJERO
Descanso	Invierno	Normal	Pareja	Ciudad Campo	Adultos Ancianos	6. PLAYA-CANARIAS
Turismo	Primavera-otoño	Normal	Pareja	X	Adultos	7. GRANDES CIUDADES
Turismo	X	Superior	Pareja	X	Adultos	8. PAISES EXOTICOS
Turismo	Primavera-otoño	Normal	Pareja	X	Adultos	9. TOUR-AUTOBUS
Descanso	Primavera-otoño	Superior	Pareja	X	Adultos Ancianos	10. CRUCERO-MAR

Figura 2.—Matriz de relación Opciones del usuario - Soluciones propuestas para el ejemplo de la agencia de viajes.

Con esto le decimos que tiene que determinar el viaje-propuesto y él buscará siempre llegar a ello. Una vez completada la matriz de la figura 2 se ha finalizado el análisis del conocimiento; la primera regla puede ser:

if	Motivo = Diversión		and
	Fecha-viaje = Verano		and
	Dinero = Normal		and
	Personas = Familia		and
	Lugar-resid = Ciudad		or
	Lugar-resid = Campo		and
	Edad = Adultos		
Then	viaje-propuesto = Playa-Apartamento-España.		

Para ilustrar cómo trabaja M.1 y las facilidades de comunicación con el usuario que presenta, podemos construir un prototipo con esta única regla. Primeramente, y trabajando con un IBM-PC y un procesador de textos como puede ser Wordstar o cualquier otro, creamos un fichero donde escribimos el "Goal" y la regla número 1. La forma de escritura es idéntica a como se ha realizado arriba. Cerramos el fichero y vamos al programa principal M.1, desde donde llamamos al fichero creado anteriormente.

M.1 empieza comprobando el fin (Goal). Observa si en su memoria activa consta este dato; si no lo encuentra prosigue la ejecución del programa buscando alguna regla que concluya con viaje-propuesto. Como encuentra la que le acabamos de introducir empieza a chequear las cláusulas "if" o antecedentes de dicha regla. El primer antecedente es "Motivo". Busca en su memoria de trabajo la información de ese campo; como no la encuentra mira si alguna regla concluye con un consecuente "Motivo", lo que no ocurre, ya que el fichero contiene una única regla.

M.1 resuelve el conflicto de la única manera posible: preguntando por el dato que le hace falta:

What is the value of: Motivo?
(¿Cuál es el valor de "Motivo"?)

Se puede contestar lo que se desee, pero si la respuesta no es "Diversión" recibiremos un mensaje de M.1 diciendo:

viaje-propuesto was songht, but no value was concluded.
(he buscado la solución pero no he encontrado ninguna).

Si se contesta con "Diversión" y a todos los otros campos correctamente la solución será:

viaje-propuesto=Playa—Apartamento—España.

Este sencillo programa cualquier persona que conozca algún lenguaje de programación lo podría realizar fácilmente, pero le sería mucho más difícil crear uno que pudiera trabajar con cualquier número de reglas, hiciera las preguntas apropiadas y llegara a la solución correctamente.

El prototipo completo se compondrá de las siguientes reglas, algunas de las cuales se incorporan para hacer al sistema más atractivo de cara al usuario:

regla 1	if	Motivo = Tranquilidad	or
		Motivo = Paz	or
		Motivo = Sosiego	or
		Motivo = Descansar	or
	then	Motivo = Descanso	
regla 2	if	Motivo = Esquiar	or
		Motivo = Jugar al tenis	
	then	Motivo = Deporte	

Estas reglas permiten al usuario responder de forma más natural a las preguntas del ordenador, que, por otra parte, se pueden estructurar de forma distinta a la que llevaba inicialmente; así cuando le falte el dato del factor "Motivo" preguntará

Cuestión (Motivo) = "¿Cual es el motivo de sus vacaciones?"

De igual forma el usuario dispone en todo momento de las posibles respuestas a cada pregunta:

legalvals (motivo) = tranquilidad, paz, sosiego, descansar, descanso, esquiar, jugar al tenis, deporte, diversión, turismo.

regla 3	if	Fecha = Otoño	
		Fecha = Primavera	
	then	Fecha-viaje = Primavera-Otoño	
regla 4	if	Fecha = Junio	or
		Fecha = Julio	or
		Fecha = Agosto	or
		Fecha = Septiembre	
	then	Fecha-viaje = Invierno	
regla 5	if	Fecha = Dic	or
		Fecha = Enero	or
		Fecha = Febrero	
	then	Fecha-viaje = Invierno	
regla 6	if	Gastos =75.000 pts/persona	
	then	Dinero = Normal	

regla 7	if	Gastos>75.000 pts/persona	
	then	Dinero = Superior	
regla 8	if	Edades<30	
	then	Edad = Joven	
regla 9	if	Edades>30	and
		Edades<60	
	then	Edad = Adulto	
regla 10	if	Edades>60	
	then	Edad = Anciano	
regla 11	if	Motivo = Diversión	and
		Fecha-viaje = Verano	and
		Dinero = Normal	and
		Personas = Familia	and
		Lugar-res = Campo	or
		Lugar-res = Ciudad	and
		Edad = Adultos	
	then	Viaje-propuesto = Playa-Apartamento-España	
regla 12	if	Motivo = Diversión	and
		Fecha-viaje = Verano	and
		Dinero = Superior	and
		Personas = Pareja	and
		Lugar-res = Campo	or
		Lugar-res = Ciudad	and
		Edad = Adultos	
	then	Viaje-propuesto = Playa-Hotel-España	
regla 13	if	Motivo = Descanso	and
		Fecha-viaje = Verano	or
		Fecha-viaje = Invierno	and
		Dinero = Normal	and
		Personas = Pareja	or
		Personas = Grupo	and
		Lugar-resid = Ciudad	and
		Edad = Jóvenes	or
		Edad = Adultos	
	then	Viaje-propuesto = Montaña-España	
regla 14	if	Motivo = Deporte	and
		Fecha-viaje = Invierno	and
		Dinero = Normal	and
		Personas = Pareja	or
		Personas = Grupo	and
		Lugar-res = Ciudad	or
		Lugar-res = Costa	and
		Edad = Jóvenes	
	then	Viaje-propuesto = Esquí-España	
regla 15	if	Motivo = Deporte	and
		Fecha-viaje = Invierno	and
		Dinero = Superior	and
		Personas = Grupo	and
		Lugar-res = Ciudad	or
		Lugar resid = Costa	and
		Edad = Jóvenes	

	then	Viaje-propuesto = Esquí-Extranjero	
regla 16	if	Motivo = Descanso	and
		Fecha-viaje = Invierno	and
		Dinero = Normal	and
		Personas = Pareja	and
		Lugar-res = Ciudad	or
		Lugar-res = Campo	and
		Edad = Ancianos	or
		Edad = Adultos	
	then	Viaje-propuesto = Playa-Canarias	
regla 17	if	Motivo = Turismo	and
		Fecha-viaje = Pri-Oto	and
		Dinero = Normal	and
		Personas = Pareja	and
		Edad = Adultos	
	then	Viaje-propuesto = Visita-grandes-ciudades	
regla 18	if	Motivo = Turismo	and
		Dinero = Superior	and
		Personas = Pareja	and
		Edad = Adultos	
	then	Viaje-propuesto = Países-exóticos	
regla 19	if	Motivo = Turismo	and
		Fecha-viaje = Pri-Oto	and
		Dinero = Normal	and
		Personas = Pareja	and
		Edad = Adultos	
	then	Viaje-propuesto = Tour-autobús	
regla 20	if	Motivo = Descanso	and
		Fecha-viaje = Pri-Oto	and
		Dinero = Superior	and
		Personas = Pareja	and
		Edad = Adultos	or
		Edad = Ancianos	
	then	Viaje-propuesto = Crucero-mar	

Paso 4: Desarrollo con el sistema concha de un prototipo

Llegados a este punto sólo tenemos que escribir la base de conocimientos anterior y el sistema concha M.1 se encarga del resto: maneja las reglas y elige las que tiene que aplicar en cada caso, pregunta cuando desconoce algún factor y es capaz de mostrarnos cómo ha llegado a esa solución o por qué, en un momento dado, hace una pregunta determinada.

Estas dos últimas características son de gran ayuda e importancia para el encargado de poner a punto el programa, facilitándole la depuración y eliminación de errores del sistema.

Paso 5: Ampliación y modificación

La facilidad de ampliar una base de conocimientos debe ser una de las características más importantes de los sistemas conchas. En nuestro caso se pueden escribir hasta 200 reglas en la base de M.1, con lo cual se podrían completar todos los casos que no se han introducido en el prototipo.

En el ejemplo de la agencia de viajes se ha concedido un factor de importancia, o de verosimilitud, al resultado de cada regla de 1.0, lo que no es en todos los casos cierto; se deberán introducir en cada caso los factores de certeza correspondientes.

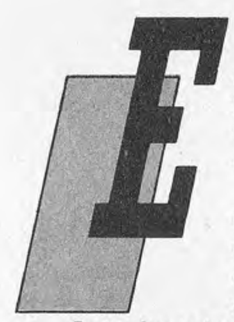
Paso 6: Puesta al día y revisión

Es sólo una extensión del paso 5 y consiste en adaptar la base de conocimiento a las situaciones cambiantes que concurren en el sistema experto.

Aunque el ejemplo era realmente simple, con él nos hemos podido dar una idea del sistema utilizado en estos pequeños sistemas basados en el conocimiento.

CAPITULO VIII

FUTURO DE LOS SISTEMAS EXPERTOS



Es muy difícil intentar describir cuál va a ser el futuro de una rama de la ciencia que se caracteriza por su constante evolución y por la facilidad con que cambian sus fundamentos.

Sin embargo, aunque sea comentando grandes tendencias, sí que se puede orientar hacia donde apuntan los proyectos de investigación que se realizan en la actualidad y que, a buen seguro, representan el futuro de todo el sector.

Actualmente en Estados Unidos se han definido dos tendencias muy diferentes sobre cómo interpretar la filosofía de utilización y desarrollo de los sistemas expertos.

Por un lado se encuentran los defensores del formalismo y la lógica como pilares de su desarrollo.

Son científicos que pretenden comprender el mundo mediante la lógica. Intentan formalizar todo lo que ocurre en la vida. Los límites de los sistemas expertos, en este sentido, se hallarán donde la lógica resulte insuficiente para representar la realidad. Es decir: si se pudiera representar matemáticamente el hecho "el lector se está comiendo un flan mientras lee el libro", éste podría ser introducido en el ordenador y, por tanto, manejado como información conocida. El paradigma de estos científicos podría ser: "TODO COMIENZO DE ESTUDIO DEBE EMPEZAR POR LA SINTAXIS".

Esta tendencia se encuentra ubicada en las universidades de la costa Oeste Americana como Stanford.

Por otro lado se encuentran los científicos que propugnan comenzar "entendiendo" el mundo antes de intentar formalizarlo. Ellos opinan que aunque resulte muy complicado entender los procesos, una vez conseguida esta comprensión es más fácil plasmar las relaciones en un lenguaje formal.

Para estos pensadores lo importante es el significado, la SEMANTICA, antes que la sintaxis. Así, un paso anterior al intentar representar que "el lector se está comiendo un flan mientras lee el libro" sería comprender realmente lo que esto significa e intentar introducir esa información en la representación utilizada.

Esta tendencia se encuentra sobre todo en las universidades de la costa Este Americana, como, por ejemplo, el M.I.T. (MASSACHUSETTS INSTITUTE OF TECHNOLOGY) o YALE.

Como paradigma se podría concluir: "LO QUE INTERESA SON LOS CONCEPTOS".

A la hora de condensar las tendencias, Tom Alexander las define como los «zarrapastrosos» (este) y los «pulcros» (oeste).

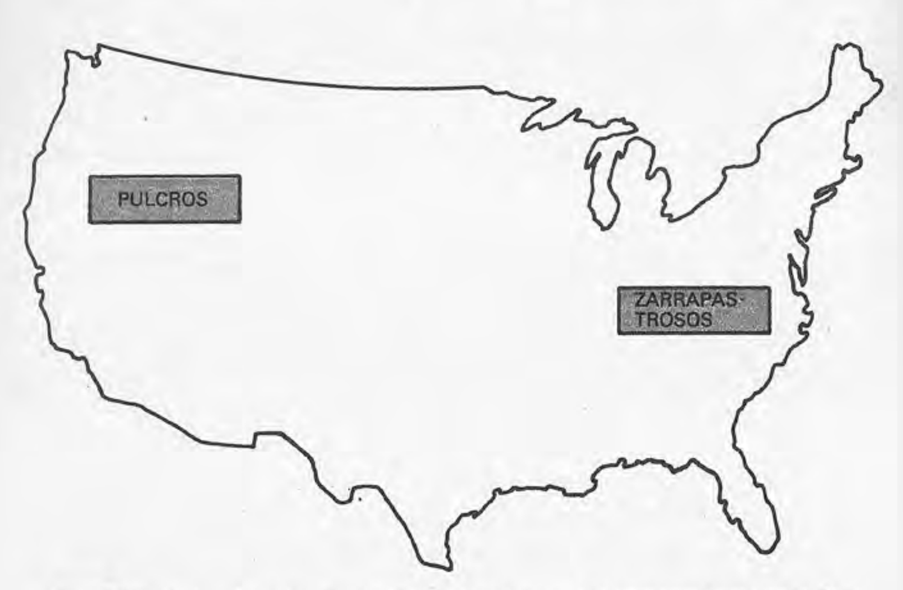


Figura 1.—Asentamiento de las corrientes de pensamiento americanas en lo que respecta a los sistemas expertos.

La quinta generación de ordenadores

Referido al futuro de los sistemas expertos habría que comentar, sin falta, el famoso proyecto QUINTA GENERACION DE ORDENADORES.

En octubre de 1981 Japón anunció que, mediante la creación de un proyecto informático que tendría una duración de diez años, pretendía crear un nuevo modelo de ordenador que consiguiera revolucionar el mundo del tratamiento de la información.

Pese a que no se han conseguido todavía todos los resultados esperados, el proyecto está obteniendo unos avances significativos en ese área y, además, ha logrado que otros países se interesen lo suficiente por el tema como para empezar sus propias investigaciones.

Analicemos brevemente lo que se pretende conseguir en el proyecto de quinta generación, paso a paso.

- RAPIDEZ. Se pretende conseguir que los ordenadores funcionen más rápidamente, es decir, que sean capaces de procesar más instrucciones e información que los ordenadores actuales en el mismo tiempo.

  Esto se intenta mediante unos CHIPS más rápidos con capacidad de proceso, es decir, de realizar operaciones básicas.
- TRABAJO CON LOGICA SIMBOLICA. Actualmente, los ordenadores funcionan con estructuras fundamentalmente numéricas. Aunque también manejan símbolos distintos de los números, todavía resulta algo farragosa la utilización de letras y caracteres no numéricos en programas de todo tipo.

  Se pretende, con este impulso, crear un software (estructuras de programas, programas en general) capaz de permitir manejar en el futuro cualquier tipo de información de una manera óptima.

  Esto permitiría una mejora sustancial en el proceso del conocimiento, tema que está directamente relacionado con los sistemas expertos.
- UTILIZACION DEL LENGUAJE NATURAL. Desde luego, esta posibilidad no se conseguirá en un 100% hasta dentro de muchos años. Sin embargo, muchos tipos limitados de lenguaje natural se pueden conseguir sin demasiados problemas.

  El proyecto de ordenadores de la quinta generación propone la consecución de esta característica tanto en concepto de comprensión del usuario (que el ordenador sea capaz de comprender lo que el usuario le mande en un lenguaje común), como en generación de mensajes (que el ordenador sea capaz de generar la información que suministra al usuario utilizando el lenguaje del propio usuario) incluso de viva voz, es decir, que el ordenador sea capaz de hablar.
- ARQUITECTURA TIPO PARALELO. Hasta ahora los ordenadores trabajan de una forma secuencial, es decir, el microprocesador realiza las tareas asignadas una por una.

En el plan de ordenadores de la quinta generación se pretende conseguir que existan muchos microprocesadores que realicen muchas tareas a la vez, pero sin que dependan siempre de un mismo "jefe". Esto resulta muy complicado y costará también mucho el conseguirlo. En cualquier caso, el proyecto está suponiendo y supondrá un gran avance en la técnica informática.

CAPITULO IX

REFLEXIONES FINALES

A menudo el escollo más importante que debe superar una nueva tecnología para su implantación se encuentra en la propia sociedad humana. El hombre necesita tiempo para adaptarse a cambios que varían sus costumbres, y más si éstos vienen a cuenta de "el futuro". De por sí, la informática ha supuesto una transformación tan radical de la sociedad humana que para muchas personas está siendo imposible seguir el ritmo.

Como parte integrante de esta revolución informática se encuentra la Inteligencia Artificial. De reciente conocimiento a nivel general se encuentra en estos momentos en plena "digestión social".

Mientras que en el mundo de la técnica y la gestión se ha acogido con curiosidad, muchos otros presentan ya los primeros síntomas de preocupación. Preguntas como ¿Adónde vamos a llegar? están a la orden del día.

Pero realmente: ¿Sabemos adónde vamos a llegar? ¿Tenemos seguridad de que el problema no se nos escapará de las manos? Si se plantea la cuestión con un poco de visión futurista ¿nos asustaremos?

Podemos tener la seguridad de que los procesos industriales, tarde o temprano (y probablemente más temprano que tarde), se encontrarán totalmente automatizados. Esto quiere decir que no se necesitará personal humano para realizarlos ni para controlarlos. Conseguir este antiguo sueño humano (que trabajen otros) plantea dos grandes problemas:

- ¿Puede ocurrir que un "error" a gran escala pueda llegar a paralizar la actividad mundial? Resulta claro el considerar este problema como ficticio, precisamente por la enorme capacidad de operación y trabajo autónomo que pueden conseguir los ordenadores.
- ¿Será capaz la sociedad de aceptar un cambio radical en sus estructuras y la llegada de "la sociedad del ocio"? Tal vez sea el proceso más difícil de alcanzar por su complicación.

  El ser humano debe asumir lo que en un principio resulta fácil: hay que transformar el hombre-trabajador en hombre-cultural.

Asimismo, inherente a todo este proceso, se encuentra otro gran riesgo. Tal y como se está llevando el desarrollo informático actualmente, de manera anárquica, mucha gente se plantea el problema de mantener intacta la libertad de operación, actuación y pensamiento que exigen los derechos naturales del hombre. Realmente la impresión general existente sobre la informatización es una "pérdida de libertad". Esto puede resultar engañoso por cuanto lo que se critica puede ser el "no poder hacer lo que no debo".

El planteamiento anterior, la pérdida de libertad, tiene sus raíces en la siguiente pregunta: ¿Puede un ordenador llegar a decidir de forma global sobre algo? El problema de los temas de decisión es importante. Nuestro mundo se vanagloria de ordenadores con mayor rapidez, capacidad de raciocinio y decisión, tal vez sin sopesar la necesidad de controlar su evolución. Se necesitarían unas reglas básicas de la informática, al estilo de las de robótica propuestas por Isaac Asimov (*).

En cualquier caso no es motivo de gran preocupación, sino de establecer un simple control. Mientras tanto, la evolución continúa y serán los hombres que estudien estos temas los que tendrán en sus manos el poder de decidir sobre toda la sociedad.

(*) Conocido escritor de obras de divulgación científica e histórica y de novelas de ciencia-ficción. En estas últimas ha tocado muy a menudo el tema de los robots inteligentes, desarrollando las que se denominó "Leyes de la robótica" (ver, por ejemplo, "Yo robot" en la colección Nebulae de Edhasa), que han sido adoptadas y utilizadas, incluso, por otros autores del género. Estas tres leyes son:

1. Un robot no puede dañar a un ser humano, ni por su inacción permitir que sufra daño.
2. Un robot debe obedecer las órdenes que le son dadas por un ser humano, excepto cuando estas órdenes están en oposición con la primera ley.
3. Un robot debe proteger su propia existencia hasta donde esta protección no entre en conflicto con la primera o segunda ley.

Aunque no sea éste el motivo más importante para muchos, cuantos más españoles trabajen sobre el tema más se nos podrá oír en el futuro. Por eso hay que animar a todos los interesados a que investiguen, estudien y propongan sus soluciones. Tanto en Inteligencia Artificial como en otras ramas en ellos se encuentra nuestro futuro.

BIBLIOGRAFIA

Sistemas expertos.
Aty, Coombs. *Díaz de Santos*. 1986.

Enseñanza asistida por ordenador.
Hudson. *Díaz de Santos*. 1986.

Construya su propio sistema experto.
Naylor. *Díaz de Santos*. 1986.

Expert Systems.
Harmon, King. *Wiley Press*.

Artificial Intelligence.
P. H. Winston. *Addison Wesley*. 1984.

Building Expert Systems.
Hayes, Waterman. *Addison Wesley*. 1985.

Introduction to Artificial Intelligence.
McDermot.

Beginner's Guide to Lisp.
Hasemer.

Programmer's Guide to Lisp.
Tracton, K.

Programming in Prolog.
Clocksin, Mellish.

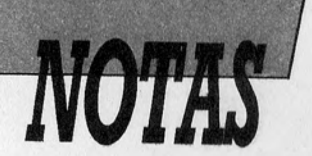
Natural Language Processing.
Grishman.

Inteligencia Artificial para la gestión de bases de datos.
Lloréns, Martín. *Fuinca*. 1986.

Entornos de Programación.
Francisco J. Garijo.

Expert Systems and Fuzzy Systems.
Nagorta.

Los ordenadores de la quinta generación.
Simons. *Díaz de Santos*. 1985.





l igual que ordenadores y paquetes de aplicación quedan rápidamente obsoletos como consecuencia de la aparición en el mercado de otros mejores en un espacio de tiempo increíblemente corto, lo mismo le ha ocurrido ya a la programación clásica: en el futuro ya no se utilizarán los programas, sino los sistemas expertos.
Los ordenadores de la quinta generación serán capaces de "hablar" sin problemas con el usuario y de programarse automáticamente a partir tan sólo de las indicaciones del usuario sobre el problema a resolver.
Ambas situaciones, que comienzan ya a ser realidad en los países más avanzados, son tan sólo algunas de los aspectos de la llamada inteligencia artificial, un nuevo reto para el hombre de nuestro siglo.

395 pts.
(incluido IVA)